Position Location of Remote Bluetooth Devices

Timothy M. Bielawa

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE
in
Electrical Engineering

Dr. Dennis G. Sweeney (Chair)
Dr. Charles W. Bostian
Dr. Timothy Pratt

June 2nd, 2005
Virginia Polytechnic Institute and State University
Blacksburg, VA

Keywords: Bluetooth, Position Location, Distance

Position Location of Remote Bluetooth Devices

Timothy M. Bielawa

Abstract

The recent proliferation of Bluetooth Devices has caught the attention of hackers. With Bluetooth devices being put in everything from cell phones to PDAs to laptops, the abuse of this technology could have an even bigger impact than the viruses and malware running rampant on the internet. Bluetooth is a short range wireless technology intended to interconnect consumer electronics devices of all kinds. The same features that make Bluetooth so attractive to manufacturers, also makes it attractive to hackers. Bluetooth devices can quickly setup up ad-hoc networks with other, previously unknown devices. Hackers have started to take advantage of the ease with which a connection can be established along with the average user's lack of computer security knowledge to break into PDAs, cell phones to steal address books and credit card numbers.

One of the largest obstacles that must be overcome in Bluetooth security is the mobility of devices and the relatively short duration of connections. In the Internet, threats can often be traced back to a source, and in many cases the source of the threat can be shut down. However, in a Bluetooth Network devices connect directly to one another, and there are no wires to follow to pinpoint the offending device. This thesis will explore the techniques for the location of Bluetooth Devices. An ideal position location system would be one that operates completely within the Bluetooth Specification. Such a system will be able to use any available Bluetooth Device to find the location of other devices. The primary focus of this thesis will be on such a system, with an overview of traditional radio position location techniques and Bluetooth so that we might gain some insight into how these techniques can be applied to Bluetooth. Data are presented from an extensive set of measurements to relate Bluetooth RSSI and distance on CSR BlueCore02 devices. Finally the results of the data are analyzed to give a rough estimate of the range error that would be incurred in the implementation of such a system.

Acknowledgements

I would like to thank my advisor, Dr. Sweeney, for all of his help and guidance during the course of my research and the preparation of this document.

I would also like to thank Tom Rondeau for all of the Bluetooth knowledge that he provided to both get me started on this project, as well as working out some of the finer details.

Finally I would like to thank all of my family and friends that have shown support along the way.

Table of Contents

List of Figures

List of Tables

1. Introduction

Bluetooth is an emerging short range wireless technology for connecting computers and peripherals. While initial adoption has been slower that predicted, many new devices are starting to incorporate Bluetooth, and Bluetooth is poised to replace many if not all of the cables running in and out of a modern computer. One of the primary adopters of Bluetooth has been cell phone manufacturers who see the technology both as a way to interface a phone and a computer without wires, but also as a way to offer a wide range of new mobile services. With a maximum range on the order of 100 meters, Bluetooth could be used to provide many short range services such as credit card replacement in point of sale equipment, or user targeted advertising where only users in specific portions of a coverage area will receive an advertisement.

Bluetooth operates in the 2.4 GHz ISM band using frequency hopping spread spectrum (FHSS) and binary Gaussian Frequency Shift Keying (GFSK) modulation. A pseudorandom hopping algorithm is used to select the next frequency out of 79 possible 1 MHz wide channels. Bluetooth devices wishing to communicate with each other form a piconet, which consists of a master, who controls all communication on the piconet, and up to 7 slaves. The slaves may only communicate with the master, and not with each other. The master and slaves will take turns transmitting, with a slave responding only when polled by the master.

Each packet has a nominal length of 625 microseconds, which defines one slot. However, a packet may be extended to fill either three or five slots. The frequency hopping pattern of the piconet is determined by the master's address, and the phase in the hopping pattern is determined by the master's clock. With this method collisions between physically overlapping piconets should be brief, while still allowing a slave to predict the channel in use by the piconet at anytime. Normally the piconet will hop frequencies between each slot, which nominally results in 1600 hops per second. When a packet fills more than one slot, the piconet will not change frequency in the middle of the packet but will wait to the end of the packet, and then hop to the frequency the piconet

would have been at had the multi-slot packet been a series of single slot packets. The Bluetooth Specification defines three classes of Bluetooth Devices, each designed with a different maximum range in mind. Class 1 devices are intended to operate at up to approximatly100 meters, while Class 3 devices are for short range application in the range of 10 meters. Class 2 devices are a somewhat ambiguously defined group that is intended for medium range applications. Each class has a defined maximum power range in which members of the class may operate. Class 3 devices may not exceed 0 dBm while operating at maximum power. Class 2 devices must output between -6 dBm and +4 dBm while operating at maximum power and Class 1 devices may not exceed +20 dBm. Any Class 1 device whose maximum power exceeds +4 dBm must implement power control, which is optional for all other devices. Using the least capable device for a desired application will reduce power consumption and can to some extent help in determining the location of a device based on which other devices can communicate with it.

As with other computer devices, the proliferation of Bluetooth devices, especially in cell phones and portable computing devices has caught the attention of hackers, and the malicious use of devices is starting to become more common. Such use often includes trying to hijack a device to steal its information in the case of a cell phone, or to introduce a virus into a portable computing device. Unlike traditional networks like the internet, it can be nearly impossible to pinpoint the location of the initial attack in a Bluetooth network, as a device could remain hidden in a room, or be carried about by the hacker. All the while the device could easily be disrupting service, over an area the size of a large auditorium of a moderate sized store.

Point of Sale applications, advertising, and the identification of malicious devices are all applications that would greatly benefit from a low cost method of being able to accurately determine the location of a remote device. An effective method should not require any specialized hardware, and should operate within the Bluetooth specifications. In the case of identifying malicious devices it may also be necessary to be able to locate a device that is not willing to explicitly cooperate in the location process. A handful of methods have

been proposed to accomplish this, however most are no more accurate than knowing whether or not a remote device is within the service volume of another know device. And not all of these methods are effective if a device is unwilling to accept connections from the position location system.

This thesis examines the feasibility of implementing a radio position location service using standard Bluetooth hardware and software. Chapter 2 presents the concepts of a two-dimensional position location system, and the traditional techniques used to provide a radio location service. It then provides a brief overview of the previous attempts at implementing a position location system using Bluetooth, including the techniques used and general accuracy of the system.

Chapter 2 examines Bluetooth in the context of the position location systems described in Chapter 3. Bluetooth timing constraints, signal strength and link quality are focused on as possible sources of position dependant data, and the theory of using signal strength to determine distance between devices is developed.

Chapter 4 presents Bluetooth power control and the issues that it presents in using Bluetooth RSSI as a method of determining received power. Possible solutions to these issues on the CSR BlueCore02 chipsets are discussed.

Chapters 5, 6, and 7 present an attempt at developing a range estimation application using the BlueCore02 chipset and standard HCI commands. Chapter 5 explains in detail the operation of the software developed to automate Bluetooth RSSI and Link Quality measurements as well as an attempt to implement the theory from Chapter 3 to estimate range between devices. Chapter 6 presents the results of the laboratory measurements made in preparation for the final range measurements. Chapter 7 presents the range measurements made, and provides an analysis of the error present in the range measurements and how they will affect the accuracy with which a device can be pinpointed.

Finally Chapter 8 provides conclusions and thoughts on the next steps to successfully estimating range between two Bluetooth devices without the use of any external hardware.

2. Radio Based Position Location Techniques

Radio based position location techniques have long been used for position location and navigation in ships and aircraft. Until the recent introduction of GPS, these systems have typically provided the user with a two-dimensional position fix constrained to the surface of the earth. This chapter will discuss the basic concepts of a two-dimensional position location system, which can then be extended to the three-dimensional case. Several systems are then presented to provide a working example of how radio techniques can be used to form a two-dimensional position location system. Finally attempted Bluetooth position location systems are presented.

2.1. Two-Dimensional Position Location Systems

Two-Dimensional Position Location systems are used to fix the location of a point in two-dimensional space. While this might be along the surface of a plane, more commonly position is fixed on the surface of the earth, or on the surface of a spheroid that represents the surface of the earth. The more common of the two-dimensional position location systems are the Rho-Theta, Theta-Theta, and Rho-Rho-Rho systems. All of these systems operate by making distance or bearing measurements on remote stations whose locations are known. Once this information is collected, it may then be combined to determine the location of the user in relation to the known stations. This relative location may then be translated into a more useful absolute location on the surface of the earth or other absolute coordinate system. Figure 1 shows graphical position location using (a) a Rho-Rho-Rho system, (b) a Theta-Theta system, and (c) a Rho-Theta system. The stars in each of the figures represent the location computed from the measurements made on each station.

2.1.1. Theta-Theta Systems

Theta-Theta systems measure the bearing from two stations to determine location. The bearings from the stations may be plotted on a map as a straight line, with the location of the receiver at the intersection of the two lines.

2.1.2. Rho-Rho-Rho Systems

Rho-Rho-Rho systems use distance measurements from three known locations to obtain a position fix. While the distances measured from a station will put the user on the surface of a sphere, Rho-Rho-Rho systems are often regarded as two-dimensional navigation systems that will place the location of an aircraft on a circle around the known station. While this is not entirely correct if the station and the user are not at the same height, as when used in aircraft, the errors introduced by not taking into account the difference in altitude of the aircraft and the ground station will be small, unless the aircraft is flying very high or very close to the station. Assuming that the distance measurements are in two dimensions greatly simplifies the calculation of position and reduces the number of ground stations required to determine the position of an aircraft.

While the distance from a single station places the aircraft on a sphere, the assumption that attitude does not matter will place the aircraft in a plane parallel to the surface of the earth. The intersection of the plane and the sphere yields a circle, on which the aircraft must be located. If measurements from two stations are taken, two intersecting circles will result, fixing the location at one of two points. Finally with a measurement from a third station any ambiguity will be resolved and the location of the aircraft can be determined.

The geometry dictates an exact solution to the problem – the intersection of the three circles result in exactly one point – and because of errors in measurements of the distances, and the lack of altitude information in the calculation, most likely, no solution will exist. That is, it is likely that a point will not exist where all three circles intersect.

The error is range measurement will result in three points of intersection, each between two circles, near the actual location of the user. The task of position location using this method is then one of determining the most likely location from the given solution. This problem is not experienced by Rho-Theta and Theta-Theta systems, where an error in a measurement will still yield a solution, which has been degraded by the error in each measurement.

2.1.3. Rho-Theta Systems

Rho-Theta systems determine location by measuring the bearing and distance from a known location. Such a system essentially defines a set of polar coordinates using the known location as the origin. The use of a VOR transmitter, described in Section 2.2, collocated with a DME transponder, described in Section 2.4, forms the ICAO's standard Rho-Theta navigation system.

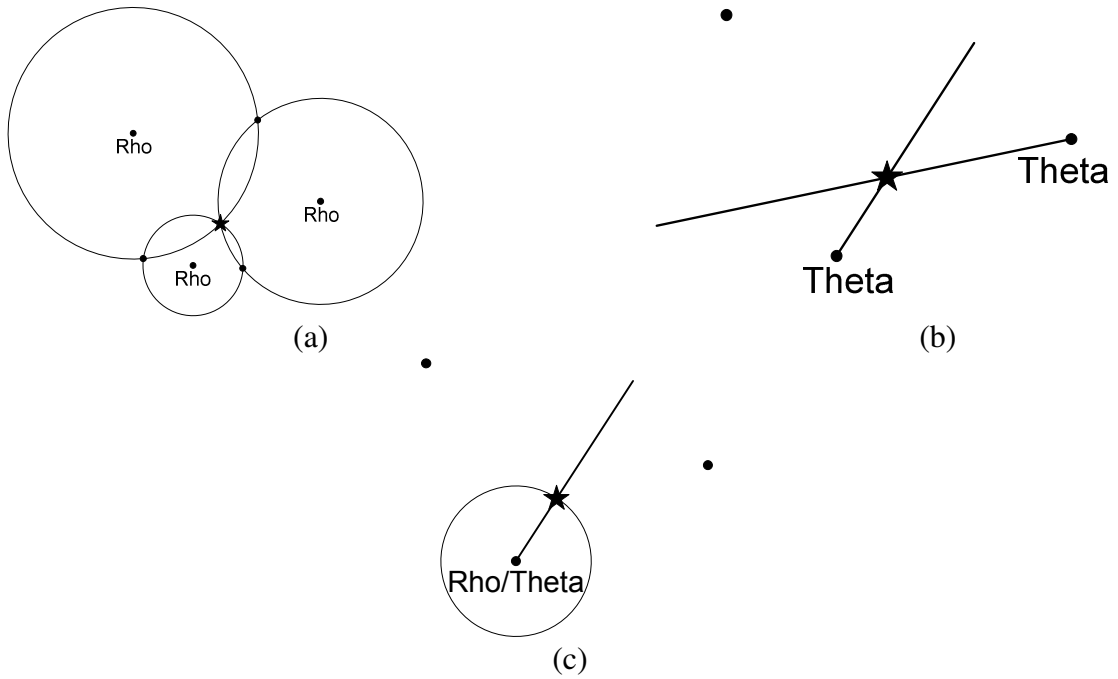


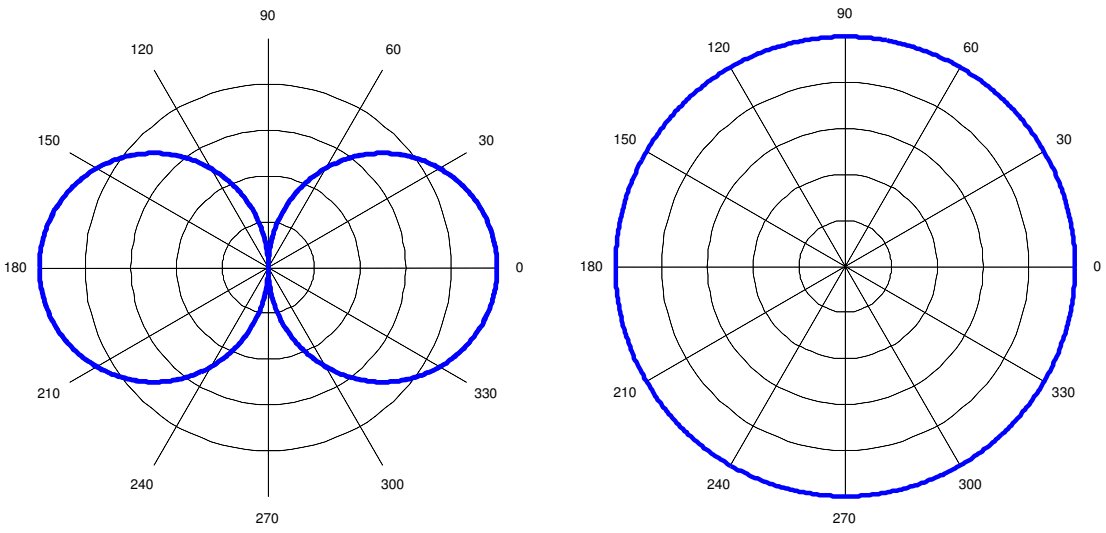
Figure 1. Position location using (a) a Rho-Rho-Rho system, (b) a Theta-Theta system and (c) Rho-Theta System. The circles in (a) and (c) indicate the range measured from the known stations, and the lines in (b) and (c) indicate the measured bearing from the known station. The star indicates the calculated position.

2.2. Doppler VHF Omni-directional Range

The Doppler VHF Omni-directional Range, more commonly referred to as a VOR, is a simple radio navigation aid in the 108-118 MHz band. Developed for, and used primarily by the aviation community, it constitutes half of the International Civil Aviation Organization's standard Rho-Theta position location and navigation system. The VOR systems consists of ground based transmitters in known locations that serve as position references, and a receiver in each aircraft that wishes to use the service. The band is split into 50 kHz wide channels, allowing for around 200 ground stations to provide coverage in the same area. The number of users of a particular station is unlimited, and the stations are unaware of the users. Once tuned to a particular station, the user may then adjust the receiver to obtain the bearing (the theta half of a Rho-Theta system) from that station, and hence the bearing to the station.

The ground station consists of 53 Alford Loop antennas, 52 of which are arranged in a circle with a 44 foot diameter, and the last antenna is placed in the center of the circle, usually elevated above the rest of the antennas to avoid interactions. The Alford Loop has the same radiation pattern, shown in Figure 2, as a vertically oriented dipole, and was chosen over the dipole for its horizontal radiation pattern. The center antenna is fed with the carrier frequency, amplitude modulated at 30 Hz. This serves as the phase reference for the system, and is the frequency that the receiver is tuned to. The carrier may also contain audio information to aide in the identification of the station. [1]

The remaining 52 antennas are typically fed one at a time with a second carrier 9960 Hz above the main carrier. They may be fed in any number of sequences to simulate the rotation of a single antenna around the circumference of the circle at 1800 revolutions per minute. The simplest sequence is to feed the antennas in order, although feeding multiple antennas at a time and in more elaborate sequences provides some advantages and are in use to some extent. While the signals fed to these antennas are unmodulated and phase coherent between the antennas, the rotation produces two effects at the receiver that are the basis of the operation of the system. [1]



(a) (b)

Figure 2. Alford Loop radiation patterns as used in a VOR transmitter: (a) in the horizontal plane, and (b) in the vertical plane. [1]

Due to the Doppler Effect, the rotation causes the second carrier to be frequency modulated at the rate of rotation, or 30 Hz, with a maximum frequency deviation of approximately 480 Hz, the number of wavelengths the beam traverses per second. Because the modulation is produced by the rotation of the beam, the phase of the second carrier varies linearly with the bearing from the station. At a location due north of the station, the modulation on the main carrier and the second carrier will be in phase, and at a location due south of the station the two will be 180 degrees out of phase. Determining bearing from the station is now a simple matter of demodulating the received signals and comparing the phase of the two. The resulting phase difference may be directly displayed to the user, and more commonly compared to a user generated reference phase to indicate deviation from desired course. Figure 3 shows a block diagram of a VOR receiver with audio capability. [1]

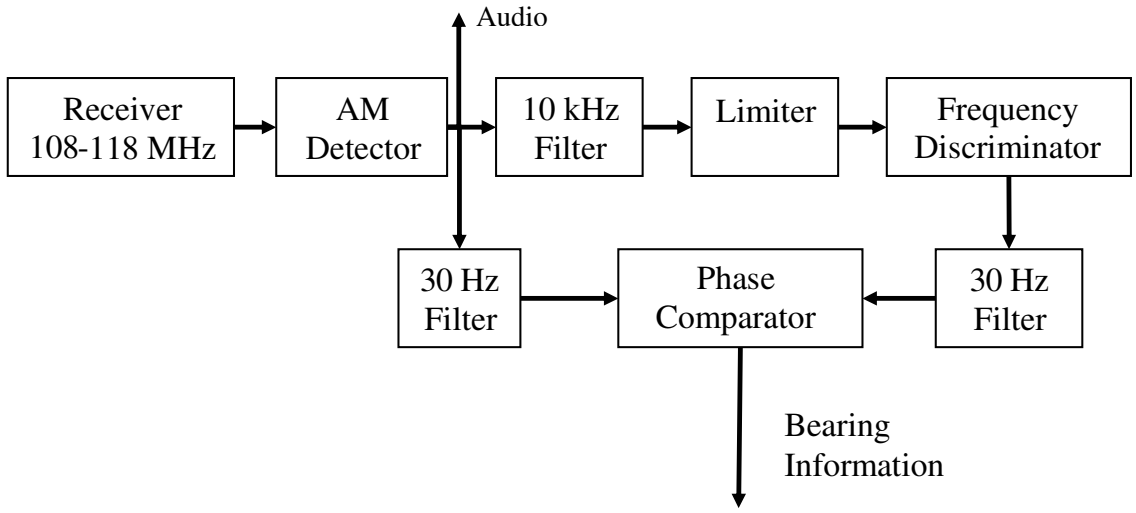


Figure 3. VOR Receiver block diagram. [1]

2.3. Automatic Direction Finder

The Automatic Direction Finder (ADF) is a system that can find the direction to a station without any intervention by the user. The direction to any station may be determined without cooperation from the station; it only needs to be transmitting long enough for the ADF to find the bearing to the station. ADF systems have typically been implemented in the 200-1600 kHz range. This allows for the use of AM Broadcast stations as well as stations specifically intended for direction finding and navigation. [1] By making bearing measurements to multiple stations, the ADF can be used to implement a Theta-Theta position location system.

Most ADF systems use a loop antenna to determine the direction of the arriving signal. The antenna pattern of the loop is a figure eight, with two sharp nulls broadside to the loop. The loop can then be rotated until the received signal strength is at its minimum to find the direction to the station. Because the loop has two nulls, one on each side, the loop alone can only be aligned with the direction to the station; there is still a 180 degree ambiguity in the direction to the station. To resolve this ambiguity a “sense” antenna can

be used. The sense antenna is an omni-directional antenna that when combined with the loop antenna produces a cardioid pattern. The cardioid pattern with its single null is suitable for making a coarse determination of the direction to that station, and then a more accurate measurement can be made using the loop alone. The nulls produced by the loop antenna are perpendicular to the single null of the cardioid so that the loop must be rotated by 90 degrees after removing the sense antenna to determine the direction to the station. Figure 4 shows the loop and sense antenna patterns as well the resulting patters when both antennas are used to resolve the ambiguity of the loop. A servo loop is often used to automate the process of searching for and holding the antenna null on the station. [2]

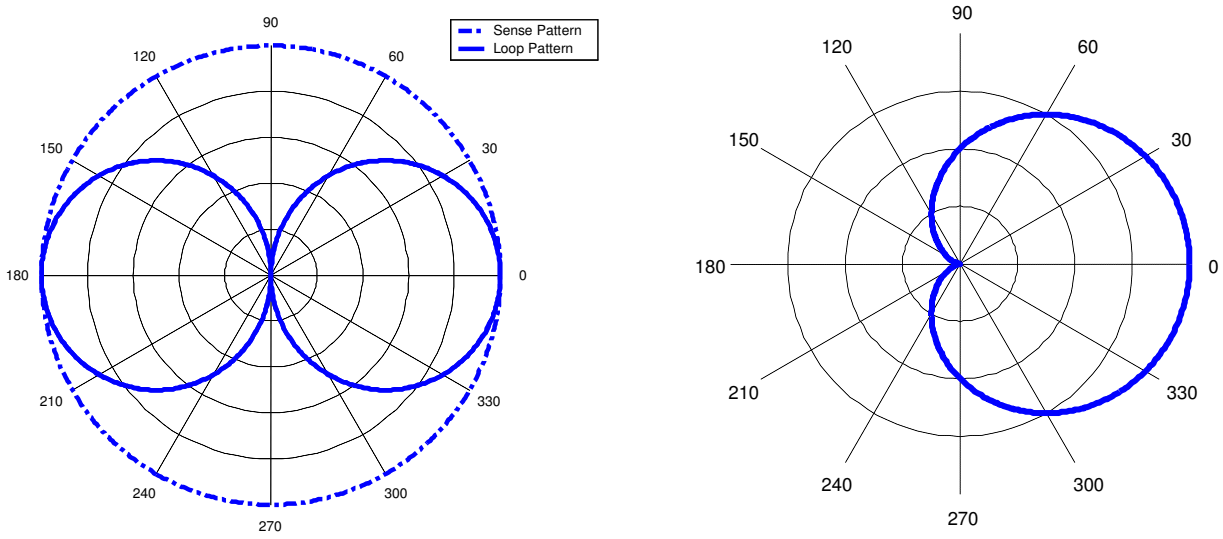


Figure 4. (a) Loop and sense antenna patterns. (b) Cardioid pattern resulting from the combination of the loop and sense antennas. [2]

Rotating the antenna itself can often be impractical. To solve this problem two loops and a goniometer may be employed. The two loops are aligned perpendicular to each other, and their outputs are connected to the goniometer. The goniometer consists of two sets of perpendicular windings and a rotor. One set of windings is connected to one loop and the other set of windings is connected to the other loop. The fields experienced by the loops are then recreated inside the goniometer, and the rotor may be turned the find the direction to the station while the antenna remains fixed. This system works just like the

movable loop setup, except that the goniometer rotor is moved instead of the loop; again a sense antenna is used to resolve the ambiguity. [2] With the antennas and the goniometer aligned properly, the shaft of the goniometer rotor may be directly used to display the bearing to the station. Figure 5 shows the goniometer and loop setup as used in a direction finding system. The sense antenna and its connection are not shown

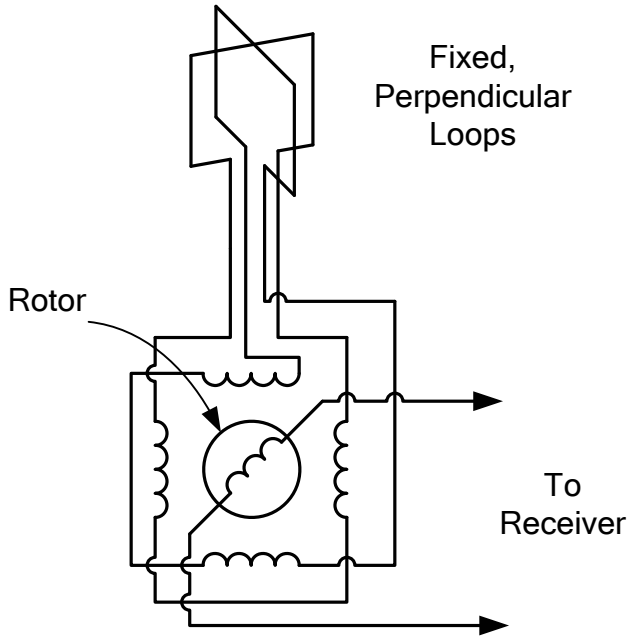


Figure 5. Goniometer and loop setup used for direction finding. [1]

2.4. Distance Measuring Equipment

As the name implies, Distance Measuring Equipment (DME) is a system that measures the distance between a ground based transponder and an air based interrogator in the 960-1215 MHz band. The system uses different 1 MHz wide channels for the interrogation and the reply, allowing for 126 ground stations serving the same area; each ground station can simultaneously support approximately 100 users. Distance is calculated by measuring the time of flight of a series of pulses between the interrogator and the

transponder. [1] DME is often used to provide the distance measurements (Rho component) of a Rho-Theta system.

When an interrogator wishes to determine the range to a transponder, it continuously transmits pairs of pulses; the pulses in each pair are separated by a fixed 12 microsecond delay. The time between pairs of pulses is long enough such that a second pair of pulses will not be sent while the first pair is still in flight at the maximum operating range of the system, 300 nautical miles. When the transponder receives a pair of pulses appropriately spaced, it waits for a fixed 50 microseconds, and then replies with two pulses, again spaced by 12 microseconds. Upon receiving the response from transponder, the interrogator may then calculate the distance between the two based on the elapsed time. [1]

Since there is no modulation on the pulses, and there is no way to tell one pulse from another, two techniques are used to prevent false detection of pulses. First the interrogation and the reply pulses are always sent in pairs spaced 12 microseconds apart. If only a single pulse is received it will be discarded. This takes care of the problem of noise spikes causing a false detection in the receiver of the interrogator. The transponder is operated in constant false alarm rate (CFAR) mode, and will always be producing responses at its maximum rate, around 3000 replies a second for a system designed to serve 100 users. By operating in CFAR mode, a transponder designed to support 100 users will always respond to the nearest 100 users, or if there are fewer than 100 users request service, will have some responses triggered by noise. This prevents the problem of adjusting receiver sensitivity under changing loads, and a constant stream of replies for the receiver in the interrogators to use in their AGCs. [1]

To prevent the receiver from mistaking replies from other users of the system as replies to its own interrogations, and miscalculating the range; the interrogator randomly varies the time between pulse pairs. This will cause all replies not triggered by the interrogator to appear to vary randomly in respect to its interrogations, while the replies that were triggered by the interrogator will remain stationary. After approximately 30

interrogations the interrogator will be able to distinguish its own replies from all of the other replies, and then it may calculate its distance from the ground station using Eq. 1. Figure 6 shows five sample oscilloscope captures from the output of a DME receiver before processing. Replies triggered by other interrogators do not line up from trace to trace, with the exception of one coincidence. The replies triggered by the interrogator appear in the same place on each trace, with the exception of the third trace where the transponder was still recovering from its last reply. [1]

$$d = \frac{t_{round-trip} - 50\mu s}{2 \cdot c} \qquad \textbf{Eq. 1}$$

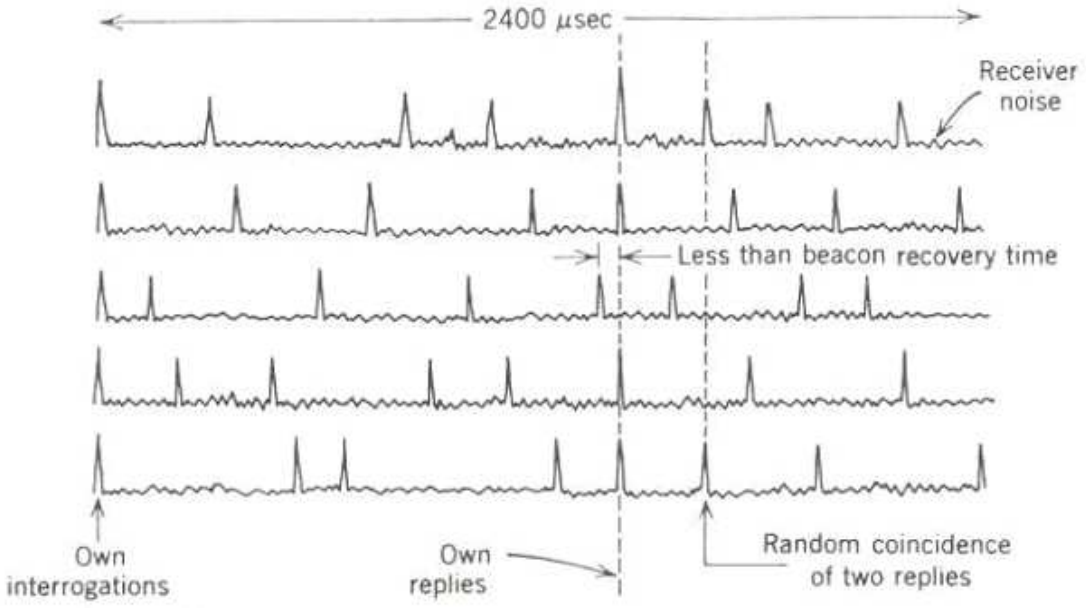


Figure 6. Sample DME Receiver output. [1]

2.5. Global Positioning System

Global Positioning System (GPS) is probably the single most common Rho-Rho-Rho 7radio navigation system in use today. Accuracies on the order of 100 meters worldwide and support for an unlimited number of users have made it popular for hundreds of

applications that were not even conceived when the system was first designed. GPS is a space based system, with all of the reference stations in orbit around the earth. One of the biggest advantages of GPS over other techniques is that once the reference stations are placed in orbit, no additional local infrastructure is needed to use the system. In contrast, systems such as VOR/DME may require dozens of ground stations every few hundred miles, which must be monitored and maintained, to provide adequate coverage. [15]

GPS consists of a constellation of 24 satellites in 6 different circular orbital planes, control and monitoring infrastructure on the ground, and the end user's receiver. Like DME, GPS calculates the distance between a station with a known location, a satellite, and the user by measuring the time of flight of a signal. After making measurements to a sufficient number of satellites, the measurements are then used to calculate the position of the user. However, unlike DME, there is no communication from the user back to the stations; the system works entirely on data broadcast from the satellites to the end users. To determine a user's position in three-dimensional space uniquely, four satellites must be visible to the user, if the user's clock is synchronized with the GPS system clock. Most receivers solve for a constrained solution in which only locations near the surface of the earth of considered. In this case four satellites are required to solve for both position in three-dimensional space, and time. The orbits of the satellites are such that there will always be at least four satellites available to a user anywhere on the surface of the earth, and there may be as many as 12 visible satellites. While additional visible satellites are not required to determine location, they may be used to increase the accuracy of the solution.

All of the satellites broadcast ranging codes and navigation data on the same frequency by using code division multiple access (CDMA). The ranging code for each satellite, which is a pseudorandom noise (PRN) code, is unique, allowing the receivers to differentiate between the visible satellites. The satellites use direct sequence spread spectrum (DSSS) techniques to spread the navigation data using the ranging codes, the result is BPSK modulated on to the carrier. Each satellite uses two ranging codes, a

Course/Acquisition Code (C/A code) and a Precise Code (P code). The C/A code has a period of one millisecond and the P code has a period of 7 days. The C/A code is available to all users, while the P code is usually encrypted and only available to military users. Military users also have access to a second frequency, which contains the same information as the civilian frequency. The use of two frequencies containing the same data allows a receiver to compensate for many atmospheric effects that will degrade system performance. [15]

Using the information in the navigation data, the receiver can determine the time at which the ranging code was transmitted by the satellite, and the location of the satellite at that time. If the receiver's clock is synchronized with the satellite's clock, then the receiver may calculate the range to the satellite by multiplying the measured propagation time by the speed of light. With range to three satellites, the receiver can calculate its location in a manner similar to that described in Section 2.1.2. The position solution from three satellites will include a single ambiguity, however that ambiguity will be above the orbit of the satellites and may be discarded for land and air based uses. The receiver's clock cannot usually be assumed to be synchronized with the system clock, and the range to a fourth satellite is required to compensate for the clock offset.

Figure 7 shows the relationship between a GPS satellite and the user in an earth centered coordinate system. The vector $\vec{s}$ points from the center of mass of the earth to the satellite, and may be computed from the navigation data broadcast by the satellite. The vector $\vec{u}$ points from the center of mass of the earth to the user's location, and is the unknown to be solved for. The vector $\vec{r}$ points from the user's location to the satellite, and its magnitude is computed from the propagation time of the broadcasts from the satellite to the user. [15] From the geometry of Figure 7:

$$\vec{r} = \vec{s} - \vec{u} \qquad \textbf{Eq. 2}$$

$$r = \|\vec{r}\| = \|\vec{s} - \vec{u}\| \qquad \textbf{Eq. 3}$$

When the GPS system clock and the receiver's clock are not synchronized the range measurements between the satellite and the receiver are know as pseudorange measurements and are denoted as ρ. Because it may not be known ahead of time if the clocks are synchronized or not, the range measurements are often referred to as pseudorange measurements even when the clocks have already been synchronized. The pseudorange between a satellite and a receiver may be thought of as the sum of the actual range, and the range error produced by the difference between the receiver clock and the system clock, t_u, as shown in Eq. 4. [15]

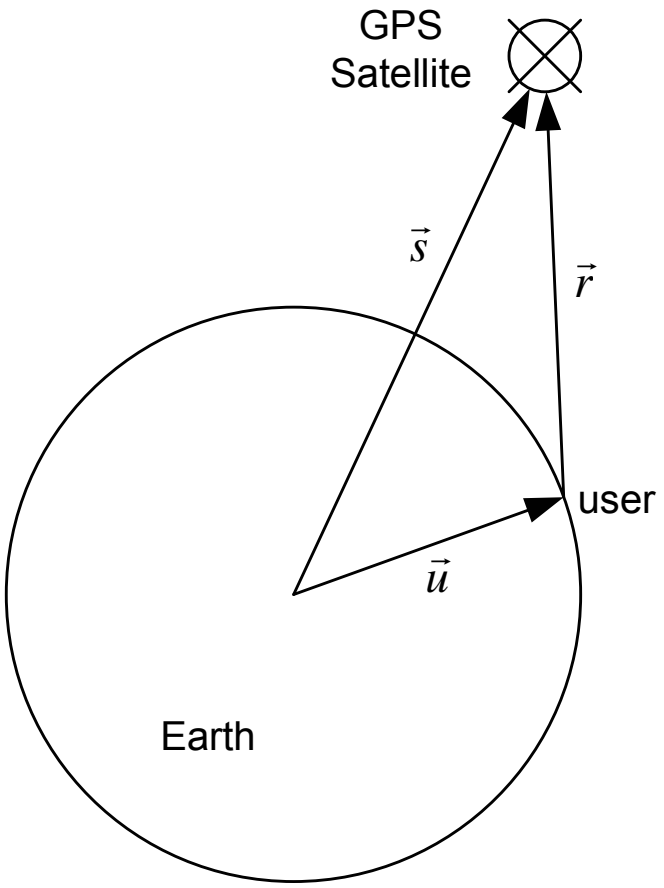


Figure 7. GPS vector representation in an earth centered coordinate system. [15]

$$\rho = r + ct_u \qquad \textbf{Eq. 4}$$

Once the pseudoranges to four satellites have been measured, the system of equations of Eq. 5 through Eq. 8 may be solved to determine the location of the user (x_u, y_u, z_u), and the receiver clock offset, t_u.

$$\rho_1 = \sqrt{(x_1 + x_u)^2 + (y_1 + y_u)^2 + (z_1 + z_u)^2} + ct_u \quad \textbf{Eq. 5}$$

$$\rho_2 = \sqrt{(x_2 + x_u)^2 + (y_2 + y_u)^2 + (z_2 + z_u)^2} + ct_u \quad \textbf{Eq. 6}$$

[15]

$$\rho_3 = \sqrt{(x_3 + x_u)^2 + (y_3 + y_u)^2 + (z_3 + z_u)^2} + ct_u \quad \textbf{Eq. 7}$$

$$\rho_4 = \sqrt{(x_4 + x_u)^2 + (y_4 + y_u)^2 + (z_4 + z_u)^2} + ct_u \quad \textbf{Eq. 8}$$

Where: (x_i, y_i, z_i) is the location of satellite *i*
(x_u, y_u, z_u) is the location of the user
t_u is the difference between the receiver's clock and the system clock: $t_r - t_s$

2.6. Personal Alarm Location System

Personal Alarm Location System (PALS) is a system developed by Dominion Wireless for determining the location of personal alarms in prisons. The system consists of a network of sensors throughout the area to be monitored that each measures the signal strength of an alarm when activated. The measurements are then sent to a central location for processing, and based on signal strength alone, the location of the alarm can be calculated to within 20 feet. From the little information that is available on the system, it is unclear whether the location of the alarm can be computed from propagation theory alone, or if they area being monitored must be mapped out beforehand to calibrate the system. While the system claimed initial success, there has been no recent mention of the use of such systems, and many prisons are now implementing location systems based the "nearest device" concept as described in Section 2.7.1. This may indicate that using received signal strength for a position location system turned out to be infeasible. [16]

2.7. Bluetooth Position Location

Several methods of finding the location of a Bluetooth device have been proposed, however most require a widely deployed infrastructure, and the cooperation of the device to be found [3]-[8]. An ideal system would require little to no infrastructure and no cooperation from the remote device. It is also desirable to be able to determine the location a device completely within the Bluetooth specification. This would allow any Bluetooth device to be used to determine the location of a remote device, without any prior preparations.

2.7.1. Nearest Device

The vast majority of the methods proposed have used standard Bluetooth hardware. The most common of these methods is to make an estimation of location based on the visibility of devices with known locations. The simplest incarnation of this technique is presented in [3] and [5]. The proposed system maintains a location server that contains the location of all access points. Anytime that a device requires location information, it can connect to an access point, and then query the location server for the location of the access point. While this approach is simple and easy to implement, it is not very accurate on the scale of a Bluetooth device. With access points providing coverage out to 100 meters, a better solution is needed for many applications.

A slight improvement is discussed in [4]. This system uses all available Bluetooth devices, and not just access points to provide location information. It is similar to the previously mentioned systems, except that some of the devices in the system support a position location service. When a device wishes to determine its location it can search for nearby devices. Once a list of nearby devices has been created, each device in the list is asked its location directly without having to query a central position location server. If none of the visible devices support the position location service, then the central server may be queried for the location of those devices based on their Bluetooth Address. To achieve improved accuracy, the system relies on many low powered devices to form the

location network. The use of lower transmitter power levels decreases the service area of the device, thereby increasing the accuracy of the system.

An additional enhancement suggested in [4] and [6] is the simultaneous use of multiple sources to improve accuracy. It is unlikely that two devices will have the same coverage areas, and by computing the intersection of the coverage areas of all visible devices, it is possible to reduce the uncertainty in an estimation of position. Figure 8 shows the technique attempted in [4]. Even with this enhancement, the worst case accuracy is still reported to be 10 meters, which is the assumed range of a single low power device [4].

2.7.2. Link Quality and RSSI

To provide a better location solution some quality of the link between the remote device, and a device with a known location must be measured. Unfortunately the Bluetooth specification does not provide much access to the hardware layers, and provides a great deal of flexibility for manufacturers; this makes a standard implementation difficult.

The use of Bluetooth Link Quality and RSSI measurements have been suggested in [7] and [8] respectively. Both techniques map out an area to be covered prior to use. When a device wishes to determine its location at a later time, it measures the signal characteristics from all available access points, and then tries to match the measured data back to a database to determine location. The draw back to both of these methods is that the coverage area needs to be prepared ahead of time and a significant effort needs to be put into mapping the coverage area. Additionally, the Bluetooth Specifications has left the quantification of Link Quality up to individual manufacturers, so a system that works well with one particular device may not work at all with another. A more desirable system would use the measured RSSI values and knowledge of the remote device to calculate a range to the device without any mapping of the coverage area prior to use.

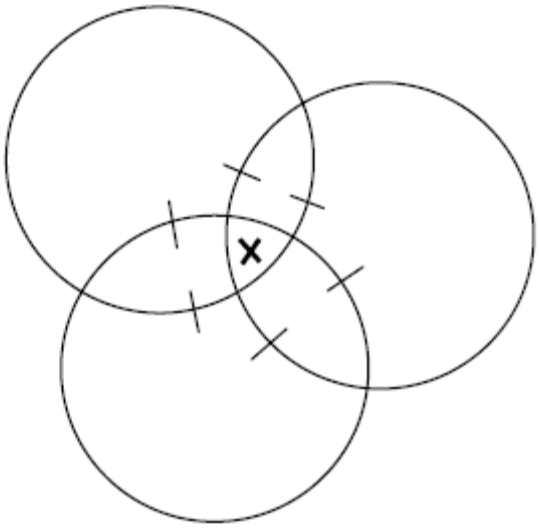


Figure 8. Graphical view of the triangulation procedure from [4]

3. Bluetooth Position Location

Bluetooth was intended to be a low cost simple solution to allow for wireless connectivity between devices. As a result of this, the specification divides the functionality of the radios in to several layers, each handling a portion of the responsibility, much like the OSI model has done for computer networks. The lower layers are typically controlled by firmware and hardware while the upper layers handle any user data, and may permit the execution of user applications. Communication between user applications, running off the device, and the lower layers, running on the device, may be performed through the Host Controller Interface (HCI) as shown in Figure 9.

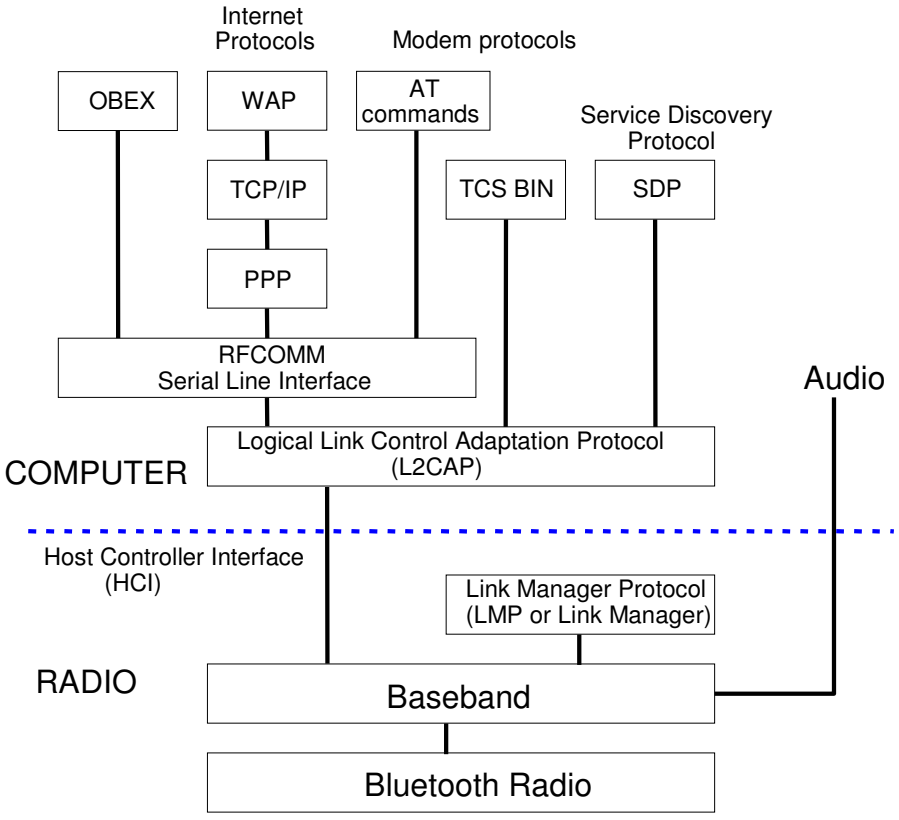


Figure 9. The Bluetooth Protocol Stack.

While such a modular approach with a standard interface provides many benefits, it makes it difficult if not impossible for the upper layers to obtain information about the operation of the lower layers that was not anticipated when the specification was written.

From Figure 9 it can be seen that the HCI is the only method of passing information from the lower layers up to the higher layers. Obtaining information from the lower layers that was no intended to be passed over the HCI would require custom low level firmware to implement custom HCI commands.

Of the information that is accessible over the HCI, timing, Link Quality and signal strength are the most promising for use in a position location system. Properly implemented, any one of these pieces of information could be used to obtain the distance between Bluetooth Devices, and from there to construct a Rho-Rho-Rho position location system.

3.1. Distance Measurement

3.1.1. Time of Flight

Bluetooth Devices wishing to communicate with each other set up a network known as a piconet. A piconet consists of a master device, and at least one, but not more than seven slave devices. The master controls the timing and the activity in the piconet, and all of the slaves must synchronize with the master. The piconet employs a time division duplex (TDD) scheme in which the master and the slaves take turns transmitting. Based on the master's clock, time slots are 625 microseconds long. The master will always start its transmission in an even numbered time slot and the slaves may only start their transmissions in the odd numbered times slots. The Bluetooth specification allows for packets which last for three and five slots in addition to single slot packets. When the slave transmits a multi-slot packet, the master will lose a chance to transmit, and normal operation will resume following the end of the slave's transmission. The master is also allowed to transmit multi-slot packets, and in this case the slave will lose a chance to transmit. Packets that occupy an even number of time slots are not allowed as they would not allow normal single slot operation to resume immediately following the multi-slot packet.

The master of each piconet is tasked as the timekeeper of the piconet. Upon receiving a packet from the master, each slave will compare its clock to the master's clock, and compute a clock offset that will allow it to synchronize its transmissions to the master's clock. To prevent collisions between slaves, and to help with clock synchronization, a slave may only respond to a master in the time slot directly following a transmission by the master. Depending upon the packet type, the preceding transmission by the master may or may not have to be addressed to a particular slave to allow a response from that slave. For most packet types, however, the master will have to specifically request a response from a slave in order for that slave to be allowed to transmit.

Unfortunately the specification allows for 10 microsecond of average clock jitter in the slave clock, and 1 microsecond of instantaneous jitter, with total jitter less than the 10 microsecond average value That is to say, if the instantaneous jitter is equal to 1 microsecond, then the average jitter must be less than 9 microsecond until the instantaneous jitter decreases. Even if the master were able to measure the exact time that a packet arrived in reference to the start of a time slot, that information would not be of much use in a time of flight calculation as the distance error introduced by the clock jitter is approximately 3 kilometers.

3.1.2. Link Quality

The Bluetooth Specification defines a Link Quality parameter that can be used as a metric to determine the quality of the link with a specific device. The specification does not define how the parameter is derived from measurable qualities of link. The definition of link quality is left up to the individual hardware manufacturers, and will probably be different for each manufacturer and possibly each model of device. It is likely that however Link Quality is defined it will degrade with distance between devices, and it may be possible to empirically determine a relationship between Link Quality reported by a device and the distance to the remote device. The results of Link Quality measurements to test this hypothesis are presented in Section 6.1.6.

3.1.3. RSSI

Most Bluetooth Devices implement a Received Signal Strength Indicator (RSSI) that allows the receiver to measure the received signal strength on a connection by connection basis. While not ideal, RSSI is the best feature of Bluetooth Devices for use in position location. It is well defined by the specification, accessible to user application through the use of standard HCI commands, and it is implemented on most devices, even when not required by the Bluetooth Specification. One draw back to using the RSSI for an unintended application is that the RSSI accuracy is not specified. Device manufacturers may provide any level of accuracy that they see fit, which could lead to some devices performing much better than others in this regard. A detailed description of Bluetooth RSSI, how it relates to actual received signal strength, and drawbacks associated with its use is presented in Section 4.1.

3.1.3.1. The Friis Transmission Formula

The Friis Transmission Formula predicts the received power when the transmitter and receiver have a line of sight path between them. This prediction is based on the transmitted power, the gain of the two antennas, the wavelength of operation, and the distance between the transmitter and the receiver. Eq. 9 is the dB form of the general Friis Transmission Formula where Pt is the transmitter power, G_t and G_r are the transmitter and receiver antenna gains respectively, λ is the operating wavelength, n is the pathloss exponent and P_r is the predicted received power. [13] The pathloss exponent controls the rate at which predicted signal power decays with distance. For free space the signal power is assumed to decay over distance via an inverse square relationship, and the hence the free space pathloss exponent is 2. Because of the wide range of frequencies over which Bluetooth devices hop, and the environments in which they typically operate, Bluetooth signal propagation is typically assumed to be free space only out to 8 meters. Beyond 8 meters the pathloss exponent of Bluetooth signals is assumed to be 3.3. [14]

$$P_r(dBm) = -20\log\left(\frac{4\pi}{\lambda}\right) - 10n\log(range) + G_t + G_r + P_t(dBm)$$ **Eq. 9**

Where: P_r is the received power, r_o is the reference distance, G_t and G_r are the transmitter and receiver antenna gains respectively, n is the path loss exponent, and P_t is the transmitter power.

The Friis Transmission Formula can also be used in reverse to predict the distance between the transmitter and receiver if the transmitter power of the remote device is known along with all of the other constants. In Bluetooth devices it is difficult to determine the exact values of all of the constants in the Friis Transmission Formula. While a device may be queried to determine its transmitter power, this value may not be accurate, as it is most likely reported from a table and not actually measured. Due to process variation in the manufacturing process individual devices could vary from the vales in the table by quite a bit. For devices with removable antennas, the antenna gain could be measured, although this may not be very easy depending on the design. Fortunately all of the constant from the Friis Transmission Formula may be lumped together and then measured through a calibration procedure. The term:

$$-20\log\left(\frac{4\pi}{\lambda}\right)$$ **Eq. 10**

could be lumped in with the rest of the constants and accounted for through calibration; however it may be calculated just as easily using the center frequency of the Bluetooth band. Bluetooth operates from 2.4 GHz to 2.4835 GHz yielding a center frequency of 2.442 GHz and a wavelength of 12.3 centimeters. Plugging that value back into Eq. 10 gives an additional constant of -40.2. Lumping the remaining constants, P_t, G_t, and G_r, together into a new constant, K_1, results in Eq. 11. Finally Eq. 11 may be rearranged and both sides raised to a power of 10 to solve for range resulting in Eq. 12. Note that the constant K in Eq. 12 is not the same constant as K_1 Eq. 11, but that: $K = 10^{K_1}$.

To solve for K in Eq. 12 several measurements will have to be made between two devices to determine the received power at various ranges. Once this data has been collected K can be solved for, and Eq. 12 can be used to predict the range between devices based on the received signal strength. Because K is solved for empirically, its value will only be valid between two specific devices and will have to be recalculated for other devices. Eq. 12 describes an exponential curve and as such an exponential regression can be used to model experimental data. The constants computed by the regression will not match those in Eq. 12 exactly as the typical form of an exponential regression is slightly different, however the result will be the same: a set of constants that can be used to estimate range from received power.

$$P_r(dBm) = -40.2 - 10n\log(range) + K_1 \qquad \textbf{Eq. 11}$$

$$range = 10^{-(P_r+40.2)/10n} K \qquad \textbf{Eq. 12}$$

Where: K is a constant representing system losses, antenna gain and transmitter power.

4. Bluetooth Power Control

The Bluetooth Specification defines three classes of devices based on the maximum output power of a device. Class 1 devices are the most powerful with a maximum transmitter power of +20 dBm. Class 2 devices are allowed operate up to +4 dBm and Class 3 devices are allowed to operate up to 0 dBm. Class 2 devices completely overlap the power ranges of Class 1 and Class 3 devices, the difference being that Class 2 devices should have a nominal transmitter power of 0 dBm, while Class 1 and Class 3 devices do not have this requirement. To reduce transmitter power consumption, power control is optional for Class 2 and Class 3, as well as for any Class 1 device with a maximum output power of +4 dBm or less (the Class 2 limit). Any Class 1 device with a maximum transmitter power of greater than +4 dBm is required to implement the power control mechanism which consists of a Received Signal Strength Indicator (RSSI), a "Golden Receive Power Range" (Golden Range), and the ability to send and receiver power control messages. The power classifications, and power control limits are shown in Figure 10. To prevent a device from overloading the receiver of another device, Class 1 devices must limit their transmitter power to Class 2 limits while searching for and establishing a connection with other devices. This requirement effectively limits the range of a Class 1 device. Although two Class 1 devices may maintain a connection out to approximately 100 meters, they can only establish a connection when they are approximately within 10 meters of each other.

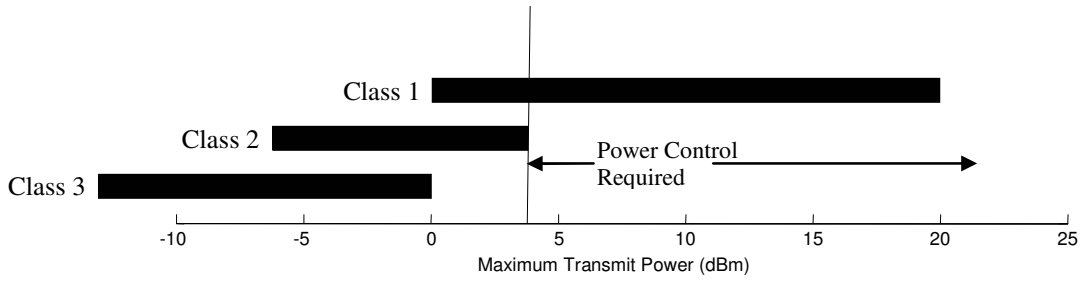


Figure 10. Bluetooth power classes and power control.

4.1. Bluetooth RSSI

Although only required of Class 1 devices exceeding +4 dBm transmitted power, most Bluetooth devices will implement power control to reduce their power consumption. One of the most important features of Bluetooth power control is the Received Signal Strength Indicator (RSSI). In addition to being used to control the output power of the transmitter on a connection by connection basis the RSSI values may be read back over the Host Controller Interface (HCI) through the use of standard Bluetooth commands.

Unfortunately the RSSI value that is returned over the HCI is not the actual received signal strength. Since the primary object of measuring RSSI in a Bluetooth device is only to facilitate transmitter power control, the RSSI measurement process has been simplified as much as possible. To help achieve this goal, the RSSI values reported over the HCI are in relation to the limits of the Golden Receiver Range as described in Section 4.1.1. While this design reduces the complexity of the power control mechanism, it also reduces the functionality of any application that may wish to use the RSSI.

4.1.1. The Golden Range

The Bluetooth Specification defines a *Golden Receive Power Range* which is a 20 dB wide window in which the receiver would like to operate. A received signal that is above (stronger than) the Golden Range will be reported as a positive RSSI value and a received signal that is below (weaker than) the Golden Range will be reported as a negative value. Any signal that falls into the Golden Range will have a reported RSSI of zero. The specifications provide a fairly loose definition of the Golden Range, so making any use of the value returned through the HCI will require some calibration between different manufacturers and probably even been different devices from the same manufacturer. The width of the Golden Range is allowed to vary by up to 6 dB and the location of the bottom of the range could vary by as much as 40 dB as shown Figure 11. Figure 12 shows the ideal relationship between the reported RSSI value and the actual received power. The curve is both linear and monotonic, with the nominal 20 dB wide Golden

Range as defined in Figure 11. Many other relationships are possible by shifting and stretching the Golden Range, and changing the curve outside of the Golden Range. The worst case is shown in Figure 13, where no information can be obtained from the reported RSSI values other than whether the received power level is above, below or in the Golden Range.

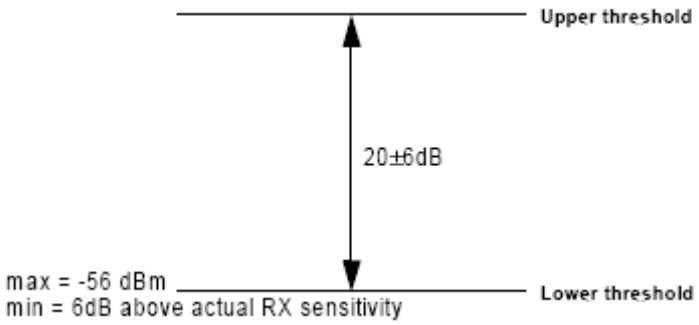


Figure 11. Definition of the Golden Range from the Bluetooth Specification.

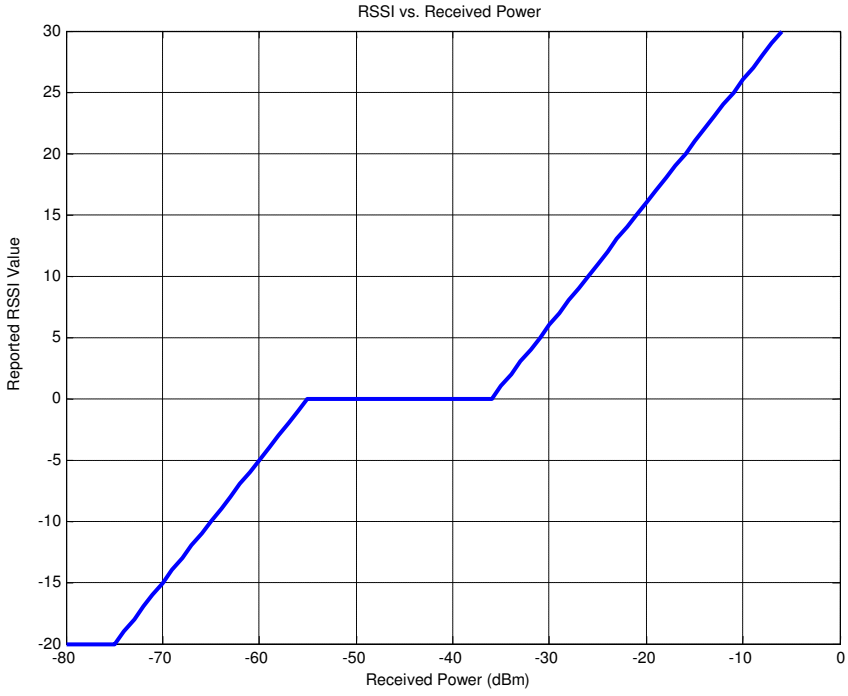


Figure 12. Ideal relationship between received power and RSSI based on the Bluetooth Specification.

4.1.2. Power Control

By defining a Golden Range, power control of a Bluetooth device becomes fairly simple. If a receiver measures an RSSI value that falls above the Golden Range (a positive value) then its Link Manager will request that the transmitter reduce its output power. If the RSSI value is below the Golden Range (a negative value), the Link Manager in the receiver will request that the transmitter increase its output power. When the transmitter changes its output power, it will do so in increments of one step for each request that it receives. If the transmitter's output is already at its maximum level, and an *incr_power_req* message is received, the transmitter's Link Manager will respond with a *max_power* message; if the transmitter's output is already at its minimum level, and a dec*_power*_req message is received, the transmitter's Link Manager will respond with a *min_power* message. Figure 14 illustrates the various LMP Power Message scenarios. Currently Bluetooth devices do not allow the receiver to request a specific step size (although this will be implemented in future revisions to the specification); they may only request an increase or a decrease. The step size may be constant, or it may be allowed to vary according to the current power level. The only requirement that the Bluetooth Specification puts on the steps is that they be between 2 dB and 8 dB.

Because the receiver cannot request a specific power step size, it does not need to know how far a received signal is outside of the Golden Range, and in fact a receiver is only required to know the sign of the RSSI value. While the Bluetooth Specification suggests that the reported RSSI value indicates how far the received signal is above or below the Golden Range in dB, simply reporting a +1 for anywhere above the Golden Range, a -1 for anywhere below the Golden Range and a 0 for anywhere inside the Golden Range is acceptable as shown in Figure 13.

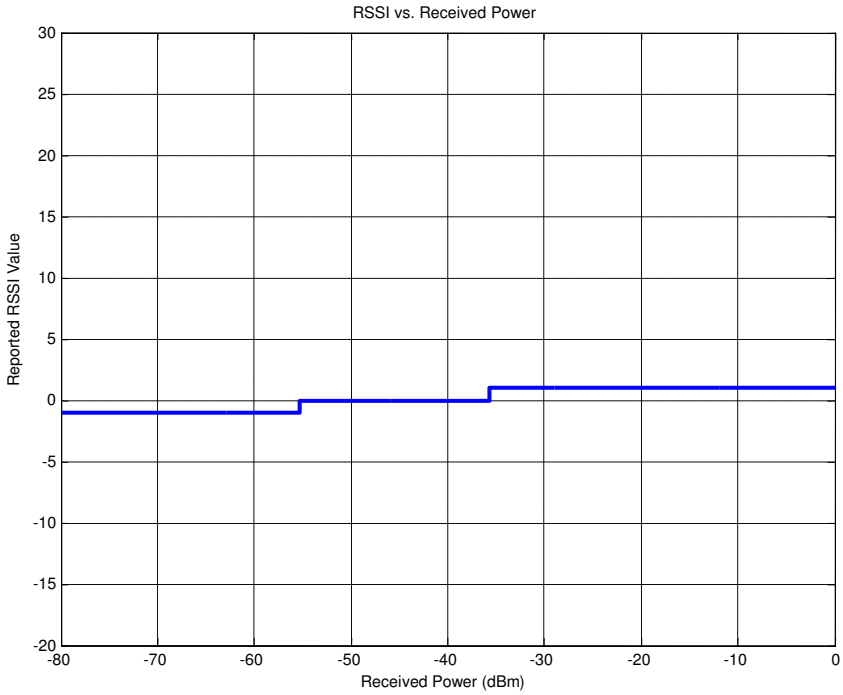


Figure 13. Worst case relationship between received power and RSSI based on the Bluetooth Specification.

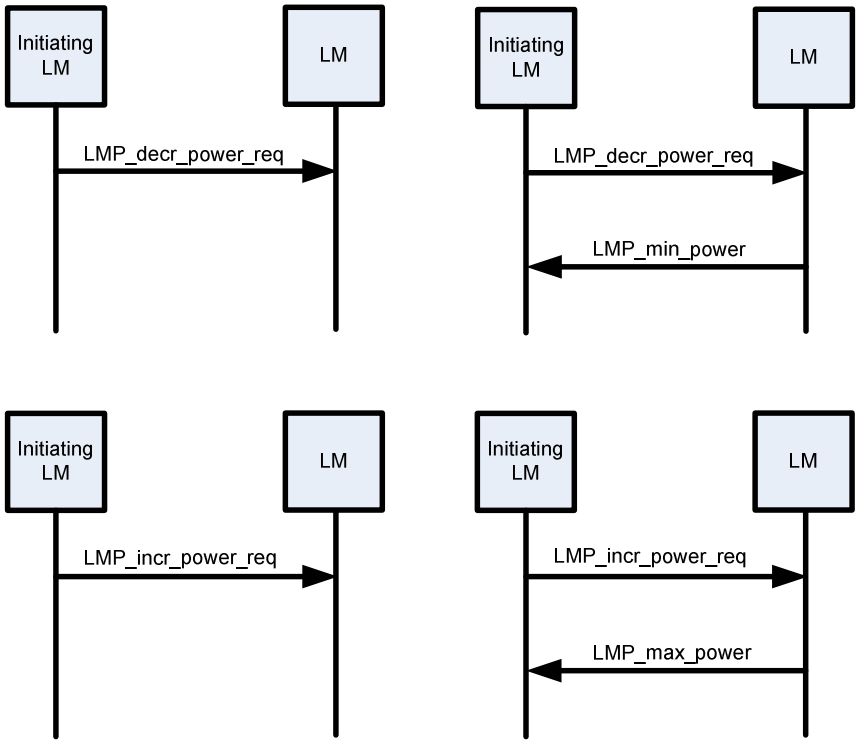


Figure 14. LMP Power Message Scenarios.

While the minimal aspects of Bluetooth power control could be implemented in hardware, a simpler and more cost effective approach would be to measure the actual received power in hardware and then use the device's firmware to create the Golden Range. In addition this would prepare existing devices for the next revision of the specification where knowing the actual received power level may be beneficial or required and it would allow for firmware adjustment to devices that originally do not meet the specification. Since knowledge of the actual received power level is not required, there is no standard HCI command the retrieve this information from devices that do measure it. Any access to this information would most likely be restricted to lower layers on the stack and require a custom implementation of the firmware stack to gain access; something that's not easily done.

Access to the lower levels of the Bluetooth Stack could also allow for additional methods of circumventing the Golden Range. By disabling or forging power control messages, the output power of the remote transmitter could be controlled in such a way as to prevent the received signal from ever falling in the Golden Range. The simplest implementation of this would be to continuously generate increase power or decrease power messages so that the remote transmitter remains at its maximum or minimum power level depending on the location of the devices. For distance measuring applications using RSSI, this is an important consideration. If the actual signal strength value cannot be obtained, then transmitter power control would have to be circumvented. With properly operating transmitter power control, a large number of the reported RSSI values would fall into the Golden Range. All of these values would be reported as zero, which does not provide any useful information to such applications.

4.2. CSR Power Control Implementation

The vast majority of the data presented in Chapters 6 and 7 was collected using the Cambridge Silicon Radio (CSR) BlueCore02 chipset. The majority of lab measurements were performed using CSR's Casira development kits, while the distance measurements were made using Uniwill and Dell Bluetooth modules incorporating the BlueCore02

chipsets. The following section provides an overview of CSR's particular implementation of Bluetooth power control for their Class 1 devices. This information was gathered from experimenting with the devices and from documents publicly available on the CSR Developer's web page.

4.2.1. RSSI and The Golden Range

The CSR documentation is not very abundant, and what is available is not very clear; nevertheless it appears that CSR devices measure the actual received power level, and then translate that value into an RSSI in relation to the Golden Range. CSR Application Note 102 [12] defines the Persistent Store Keys (PSKeys) which are various settings that CSR or an Original Equipment Manufacturer may modify to change various characteristics of a device before selling it to an end user. Some keys are used to set parameters such as the Bluetooth Address and the Device Name, and others are used for calibration of devices. Three of these keys are of interest in the context of the reported RSSI value and the Golden Range: *PSKEY_LC_ATTEN_GOLDEN_RANGE _MAXIMUM, PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM*, and *PSKEY_LC_RSSI_GOLDEN_RANGE*.

PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM, and PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM appear to define the top and the bottom of the Golden Range in terms of measured RSSI values and receiver attenuator settings. If the receiver attenuator settings go above the maximum setting a power decrease will be requested and if the attenuator settings go below the minimum a power increase will be requested. The CSR BlueCore documentation does not describe these attenuators, or how they affect the reported RSSI value or the Golden Range, other than that they exist. This is probably due to the fact that these specific Persistent Store Keys were not meant to be modified, other than by CSR

PSKEY_LC_RSSI_GOLDEN_RANGE specifies the desired RSSI value for optimal receiver operation. According to the PSKeys documentation, the default value for this key is 80, which is well outside of the range of RSSI values that can be reported over the HCI by

the CSR modules. This implies that there is a second "raw RSSI" value that the CSR devices understand which is then translated to the RSSI value as defined in the Bluetooth Specification. Again no indication is given as to how this value relates to the receiver attenuator settings or how modifying this value will affect the Golden Range. It seems most likely that this value relates an actual received signal strength to the center of the Golden Range, and that the previous two keys relate the top and bottom of the Golden Range to this value. By modifying these three PSKeys it should be possible to change the center and the width of the Golden Range, making the device more suitable for measuring the actual strength of the received signal.

4.2.2. The Power Table

The CSR BlueCore devices use a transmitter power table to control their output power. Each row in the power table contains internal and external amplifier settings for a given output power. The table is filled such that a step down one row in the table results in an increase in transmitter power, and a step up one row in the table results in a decrease in transmitter power. Table 1 is a sample power table taken from a Dell TrueMobile 300 module that uses the BlueCore chips. The first two columns of the table are the digital to analog converter values for the BlueCore's internal amplifier, and the external amplifier if present; if an external amplifier is not present, the second column must be set to zero. [11] From the Table 1 it can been seen that there is no external amplifier in the TrueMobile 300. The third column of the table is the output power the DAC settings will produce. This is the power level the device will report when queried over the HCI, and these values will be used to set the output power to an initial level. The Bluetooth Specification states that each step must be between 2 dB and 8 dB in width.

Internal Amplifier	External Amplifier	Transmit Power
2	0	-28 dBm
6	0	-24 dBm
12	0	-20 dBm
20	0	-16 dBm
25	0	-12 dBm
32	0	-8 dBm
40	0	-4 dBm
50	0	0 dBm
57	0	4 dBm
63	0	6 dBm

Table 1. Transmitter Power Table from Dell TrueMobile 300.

Using a power table instead of a fixed step size provides several benefits. Most importantly, it allows process variations to be removed during production by calibrating each device. Although all devices produced may have the same transmit power entries, the DAC entries may be different to make all of the devices operate identically. Additionally a power table allows each step size to be different based on location in the table. This could allow for larger steps at lower transmit powers and smaller steps at higher powers. A final result of the power table is that it provides a simple method of disabling transmitter power control all together or achieving results not originally intended by the Bluetooth Specification. By filling the table with incorrect values, the Bluetooth Device will not correctly report its transmitter power, and by filling all rows with the same DAC values power control will be effectively disabled.

In the latter case when a power step is made, the outputs of the DACs will remain the same, and the transmitter power will not change even though the device will think that it has changed its output power level.

5. The Range Estimation Application

Of the possible methods of determining the range between two Bluetooth Devices that were discussed, the most likely to be successfully is the correlation of RSSI and actual received signal strength. Once the actual received signal strength can be determined the range can be estimated as described in 3.1.3.1. To determine the received signal power, several RSSI reports will have to be collected and averaged. This chapter describes the software that was developed to accomplish this, the Range Estimation Application. The Range Estimation application was developed to automate the collection of both RSSI and Link Quality data, and use this data to estimate range between devices. The application was written in Microsoft Visual C++ .NET using MFC for the core application. The display portion of the application was added using National Instruments Measurement Studio. The underlying Bluetooth functionality of the application was provided by CWT's *Bluetooth Upper Layer Protocol Stack* and *Bluetooth Upper Layer Terminal Application*, which are dependant on CSR's development driver for the USB interface to the device. Figure 15 shows a block diagram of the communications flow between user applications and a Bluetooth device. Both the Range Estimation Application and the Bluetooth Upper Layer Terminal Application can communicate with Bluetooth Devices by using the Bluetooth Upper Layer Protocol Stack. The Protocol Stack can communicate with any Bluetooth Device through the RS-232 interface, and it can also communicate with devices based on the CSR BlueCore chipset through the USB interface.

The Bluetooth Upper Layer Protocol Stack provides convenient access to the functionality of the Cambridge Silicon Radio (CSR) BlueCore devices through the use of CSR's development USB driver. The stack also provides access to devices from other manufacturers through an RS-232 interface; however the stack had some issues dealing with the Zeevo BlueDolphin devices that were tested, preventing the Range Estimation Application from being used with these devices. These issues along with data collected from the BlueDolphins are discussed in Section 6.1.3.

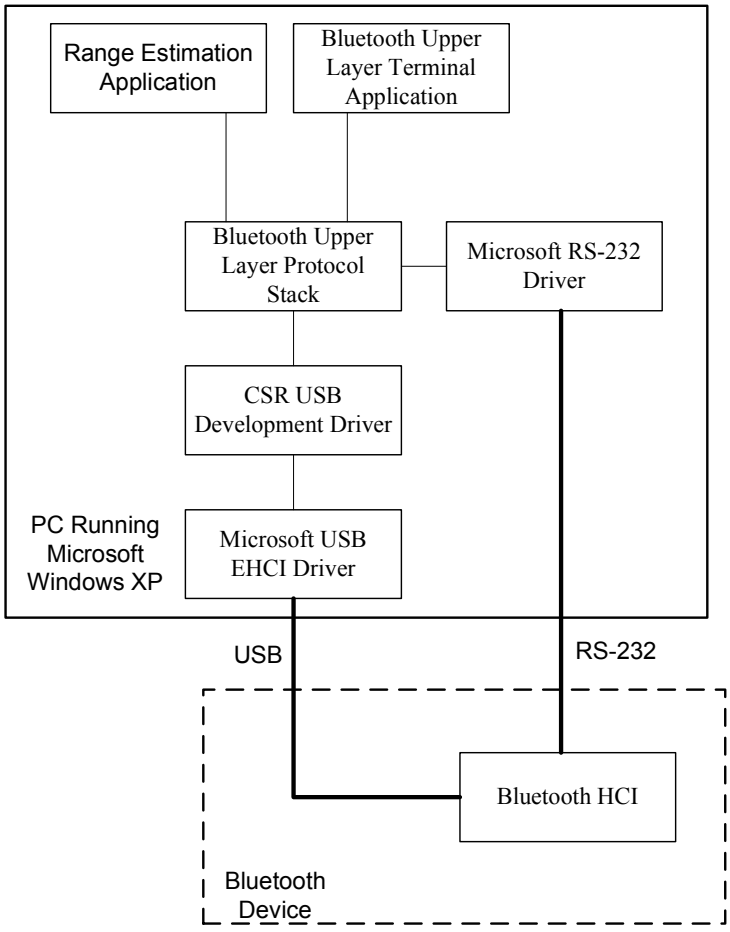


Figure 15. Block diagram showing the communications flow between the user application and the Bluetooth device. Both the Range Estimation Application and the Bluetooth Upper Layer Terminal Application can communicate with devices through the Bluetooth Upper Layer Protocol Stack. The Protocol Stack may communicate with a device over either USB or RS-232.

The Range Estimation application interfaces with the Upper Layer Protocol Stack to gain access to the Bluetooth hardware. The Upper Layer Terminal Application provides manual access to the Bluetooth device and is useful in situations where an automated approach is not practical, such as environments where maintaining a connection is difficult. By using a layered approach the Range Estimation application does not have to deal with the implementation of the CSR Bluetooth drivers, and can instead focus on collecting, processing and displaying data.

The primary function of the Range Estimation application is to collect and display RSSI and Link Quality data for a single connection with a remote device. The application

allows the user to search for and connect to a remote device, once the connection is established the application automatically starts to send packets at a nominal rate of 100 packets per second and requests both RSSI and Link Quality for the connection immediately after sending each packet. Low signal strength or high packet error rates can cause the rate of packet transmission to decrease, and the Range Estimation Application will throttle its packet rate in an attempt to maintain the maximum possible throughput in these cases.

Figure 16 shows the Range Estimation application searching for remote Bluetooth Devices; one device has been discovered so far. Searching for remote devices and connection maintenance including setup and tear down is performed from the *Output* tab of the application. The remaining tabs display the collected data in various formats. The left pane of the Output tab shows devices that have been discovered, while the right pane shows any messages returned by the *Upper Layer Protocol Stack* including RSSI and Link Quality reports and connection status. Clicking on the *Inquiry* button will cause the application to place the local Bluetooth device into Inquiry Scan and Inquiry Mode. Placing the local device in Inquiry Scan mode allows remote devices to discover the local device. If the local device is not in Inquiry Scan mode, it is not discoverable by remote devices. Placing the device in Inquiry mode causes the local device to search for other devices that are in Inquiry Scan mode. Enabling both modes at the same time, allows the application to look for and use remote devices as well as allowing remote devices to find and use the local device at the same time. While the local device may only initiate one connection at a time (an intentional limitation of the software), it can participate in any number of connections initiated by other devices. While enabling Inquiry Scan mode is not required, it allows the same application to be used for testing of remote devices as well. As remote devices are discovered, their addresses are placed in the left pane so that the user may select a device and initiate a connection with it.

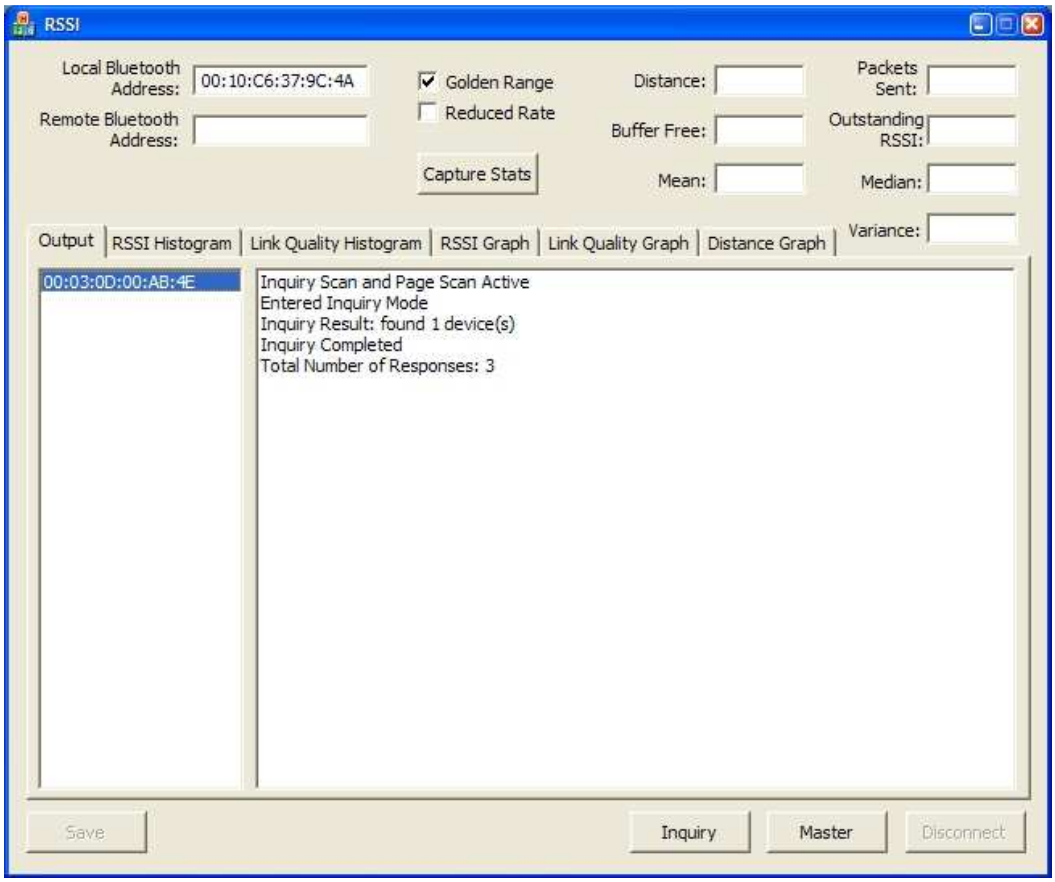


Figure 16. Bluetooth Range Estimation Application searching for remote devices, with one device discovered so far.

After finding a device in Inquiry mode, selecting the device in the left pane and clicking the *Master* button will open a connection to the device. Upon successfully establishing a connection the application will automatically start sending packets to the device, and requesting status information about the connection from the local device. Status information about the connection may not be requested from the remote device, as the Bluetooth Specification does not provide a mechanism to accomplish this. Figure 17 shows the status of a connection established with the device that was discovered in Figure 16. The right pane shows the results of the RSSI and Link Quality requests as they return from the Bluetooth Stack. This same data is also displayed graphically on subsequent tabs. The *Distance* field shows the current estimated range to the remote device in feet. The *Packets Sent* field shows the total number of packets sent to the Bluetooth Upper Layer Protocol Stack and is not necessarily the number of packets send to the remote

device; the local device may have dropped packets preventing them ever being sent. The difference in the number of packets sent to the Bluetooth Stack and the number of packets successfully transmitted, or dropped by the local device is indicated by the *Buffer Free* field. The BlueCore02 devices support an internal buffer length of eight packets. If more packets are sent to the Bluetooth Stack than can be handled by the local device, the buffer will fill up and if the application does not reduce the packet transmission rate, the buffer will overflow and the stack will crash. To prevent a stack crash, the application will stop sending packets to the stack when the buffer is nearly full. Space in the buffer will be freed up when a packet is transmitted to the remote device or dropped by the local device; however there is no method of determining which action is used to clear a specific packet from the buffer.

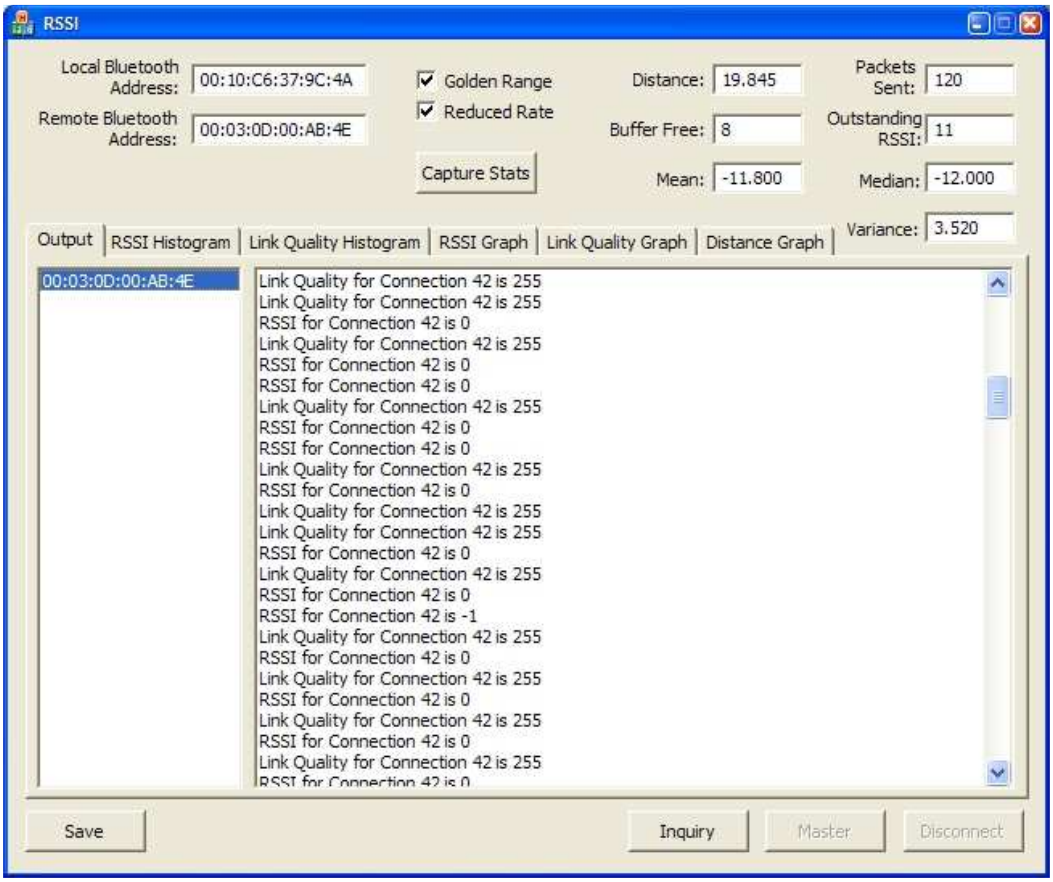


Figure 17. Bluetooth Range Estimation Application showing the text output after connecting to a remote device. Although only Link Quality and RSSI reports are seen, connection status messages will also be displayed in the right pane.

The *Oustanding RSSI* field indicates the number of packets that have been sent to the stack in excess of the number of RSSI responses from the stack. This count will include packets that the application has sent to the stack and have yet to be transmitted by the local device as well as packets that have been transmitted by the local device, but for which an RSSI request is still pending. It should be noted that a specific RSSI response can not be associated with a particular packet. Although a packet is send at the same time as RSSI and Link Quality requests are made, the rate and the order of responses is not guaranteed by the Bluetooth Stack or the Bluetooth Device. In fact a response to a specific request may never be generated at all. The effect of the lack of an order guarantee can bee seen in the right pane in Figure 17. Although all of the RSSI and Link Quality requests were made identically, the first two responses were Link Quality responses, indicating that an RSSI request was delayed.

The *Mean*, *Median*, and *Variance* fields are context sensitive and display statistics of the last 50 samples. If one of the Link Quality tabs is selected then statistics on the last 50 Link Quality samples are displayed, otherwise statistics on the last 50 RSSI samples are displayed. The *Capture Stats* button nominally collects 10 seconds worth of both RSSI and Link Quality samples and then displays the mean, median and variance of collected samples. The actual number of packets collected is based on how many packets would nominally be transmitted in 10 seconds; for the normal operation rate of 100 packets per second, the statistics of 1000 packets will be displayed. Because the application stops sending packets when the local device buffer is nearly full, it make take more than 10 seconds to collect the desired number of packets in high packet loss cases.

Checking the *Reduced Rate* checkbox will cause the packet transmission rate to be slowed by a factor of 5 to 20 packets per second. This option is useful in situations where substantial packet loss will prevent sending packets at the default rate. This option may only be changed before a connection is made as changing the data rate while a connection is active would create several display issues on the strip charts due to the inability to detect when a response to a RSSI or Link Quality request was generated.

Changing data rates during a connection would cause several packets to be displayed at incorrect times. These issues are further discussed below along with the strip charts.

The *Golden Range* checkbox determines how the application handles the RSSI data returned from the Bluetooth Device. When the box is unchecked, the RSSI values are displayed exactly as they are reported from the local device. With the box unchecked a packet with received signal strength just above the Golden Range will be displayed as +1, while a packet with received signal strength just below the Golden Range will be displayed as -1. When the box is checked, the reported RSSI values are adjusted to compensate for the Golden Range in an attempt to represent the actual received power levels. Because the Golden Range is 20 dB wide, RSSI values above the Golden Range will be shifted up by 10, and RSSI values below the Golden Range will be shifted down by 10. Values that are in the Golden Range can not be compensated for, and will be displayed as zero. By modifying the data in this way, the RSSI data now has a physical representation to it instead of just being arbitrary numbers. The modified data indicates the relationship between the received signal strength and the center of the Golden Range in dB. When a signal is just above the top of the Golden Range, the modified RSSI data will be +11, indicating the signal strength is 11 dB above the center of the Golden Range. This option can be changed while a connection is in progress, and the change will take effect immediately on new data only. There is no indication in the data as to the state of this check box, so the interpretation of the data is dependant on the state of the checkbox.

The histogram tabs of the application shows a histogram representation of the 50 most recently received RSSI and Link Quality updated in near real time. These views are particularly useful when the connection between devices is first established, or when a large change in the location of one of the devices occurs. Immediately after either event, the connection can be somewhat unstable, and the reported RSSI values will fluctuate greatly. After several seconds the RSSI histogram will stabilize, although small fluctuations will continue to occur. Figure 18 shows the RSSI Histogram several seconds after establishing a connection. The Link Quality histogram is similar; however with a

much wider range of possible Link Quality values, individual bins of the histogram are almost indiscernible.

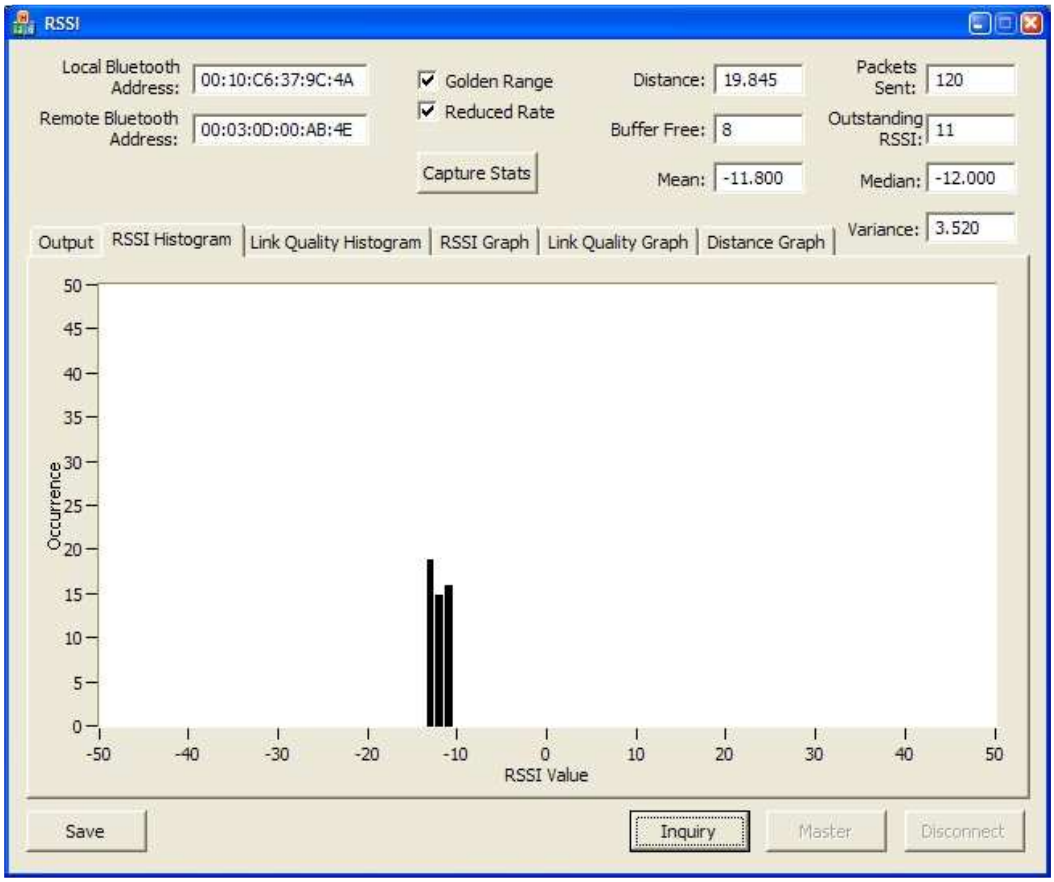


Figure 18. Bluetooth Range Estimation Application showing the RSSI Histogram after connecting to a remote device. Immediately after a connection is made the RSSI values will fluctuate by a significant amount for several seconds. By the time this screen shot was taken the RSSI values had stabilized.

The *RSSI Graph* and *Link Quality Graph* tabs show a moving strip chart of the most recently collected RSSI and Link Quality values. To attempt to display this data in a method that is independent of the selected packet transmission rate, the X-axis of both graphs is nominally 10 seconds long. Because of the lack of timing constraints from the Bluetooth Specification, and the Bluetooth Stack, it is impossible to display the actual time at which an RSSI or Link Quality measurement was made, as explained above. To get around this and still provide a useful time scale, it is assumed that all responses are timely and evenly spaced. Under these assumptions the X-axis values may be computed

by simply dividing the packet count by the packet rate. This approach works well when the connection is stable, however packet loss will cause the strip chart to progress at less than real time. While this is not ideal, significant packet loss over the course a test has rarely occurred without the connection being lost. With little packet loss, the data rate is nearly constant, and time scale is accurate. The *RSSI Graph* tab is shown in Figure 19. The *Link Quality Graph* tab is not shown as it is identical to the *RSSI Graph* tab.

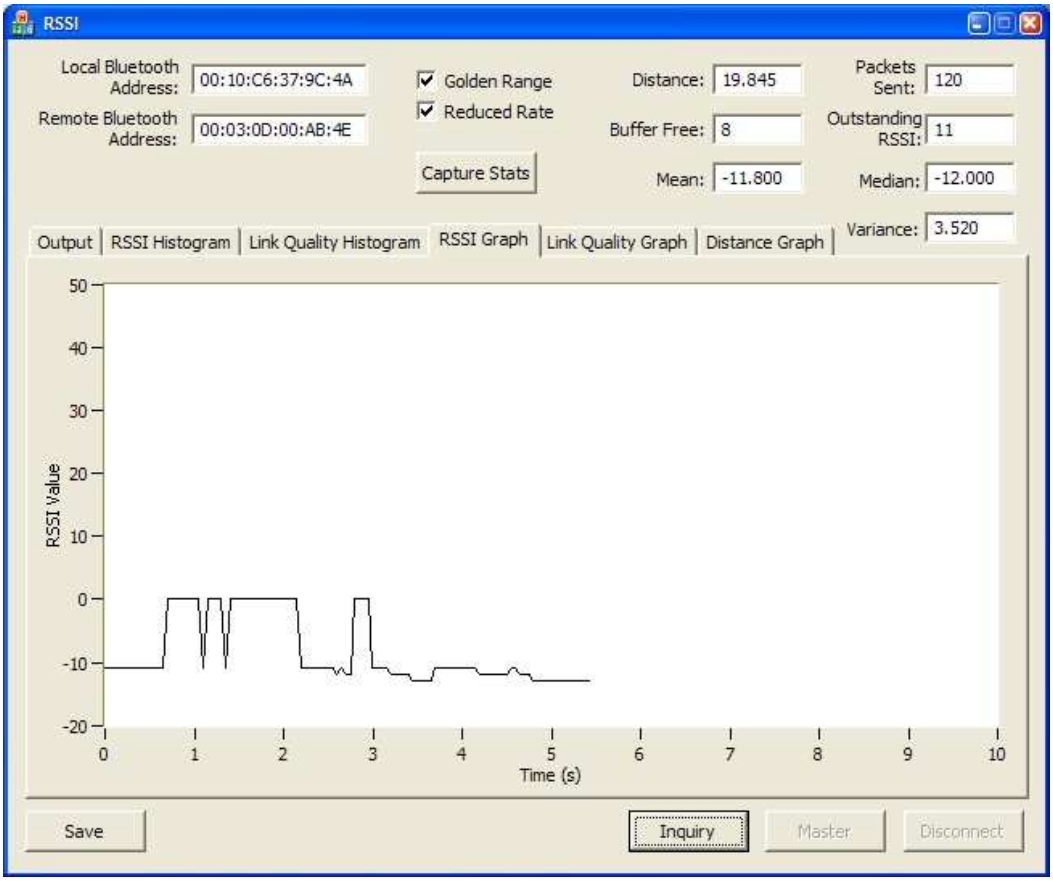


Figure 19. Bluetooth Range Estimation Application showing the RSSI graph after connecting to a remote device. The received signal strength is just at the bottom of the Golden Range, resulting in the jumps between 0 and -10.

The *Distance Graph* tab shows the most recent range estimates. This graph is similar to the RSSI and Link Quality graphs. Like the previous two graphs, it is updated four times a second. The range estimate is based on the mean of the previous 50 RSSI values, and new range estimation is computed for each RSSI value returned by the Bluetooth Stack.

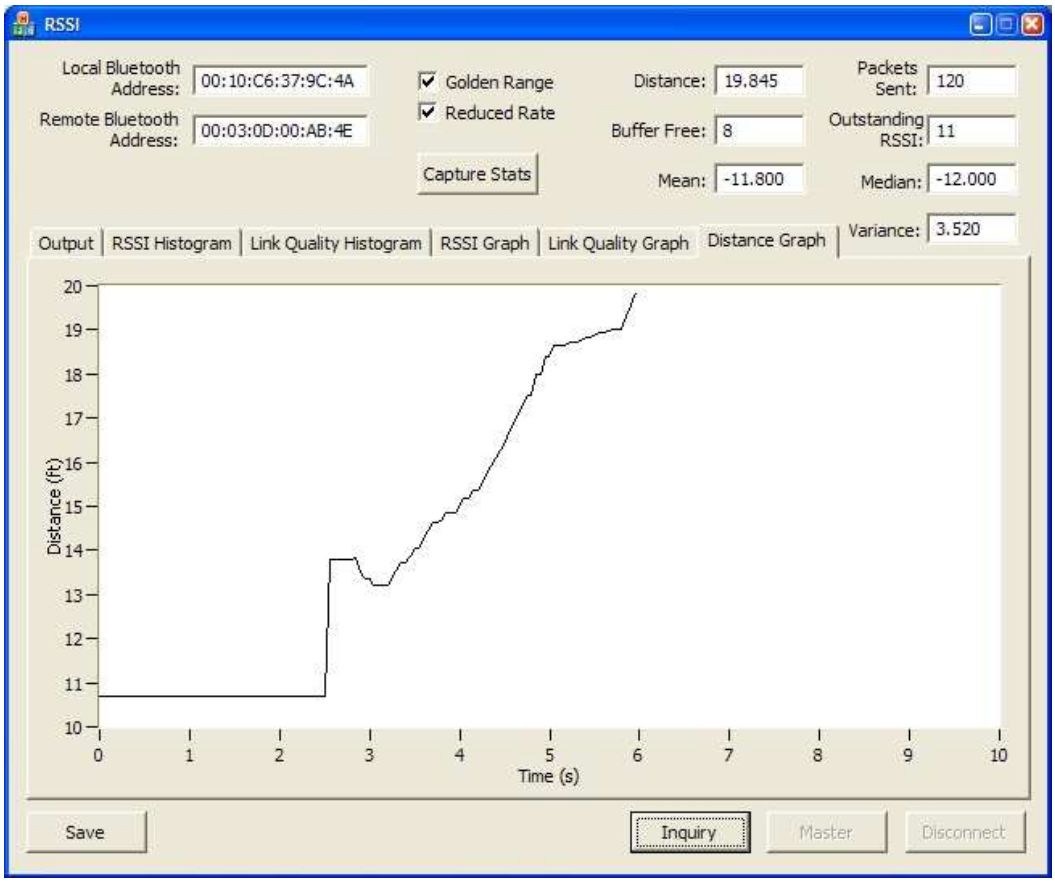


Figure 20. Bluetooth Range Estimation Application showing the estimated distance after connecting to a remote device.

As a result, the *Distance Graph* suffers the same time scale distortions as the previous two graphs when packet loss occurs. Figure 20 shows the *Distance* Graph tab. The large variation in estimated range seen in Figure 20 can be attributed to the instability seen in reported RSSI values after a connection is made. After allowing the reported RSSI values to stabilize, the range estimate will also stabilize, although large jumps will still occur if the reported RSSI values cross into the Golden Range. The estimation algorithm will see this as a change of 10 dB in received signal strength and will respond accordingly. Range estimates were originally based on the theory developed in Section 3.1.3 until sample data was collected. After a sufficient amount of data was collected the estimates were modified in an attempt to reflect the conditions seen in the data. The distance measurements used to collect this data, and the final parameters of the estimations are further discussed in Chapter 7.

6. Laboratory Measurements

Four types of measurements were conducted to examine the possibility of using RSSI and Link Quality measurements to determine range between Bluetooth Devices. The majority of the measurements made entailed connecting two devices together with a cable and a variable attenuator, and then making measurements for various attenuator settings. This setup was used for both RSSI and Link Quality measurements. The remaining measurements were actual distance measurements made by varying the distance between two devices and recording RSSI values versus distance. These measurements were conducted both indoors and outdoors.

6.1. Cable Measurement Setup

All of the cable measurements used CSR's Casira Development Kits with the majority of the measurements using two Casiras, one as the master and one as the slave device. The Casiras that were used contained a BlueCore02 module with a connectorized antenna, making it easy to connect two of the devices together using a cable. One set of measurements was performed using a Casira for one of the devices, and a Zeevo BlueDolphin as the other device. Again the BlueDolphins have a connectorized antenna making such measurements easy. Figure 21 shows the (a) outside and the (b) inside of the Casiras. The Bluetooth module is outlined by a blue clip in the middle of the Casira, with the rest of the board occupied by peripherals and test points. In this picture a small printed antenna is connected to the antenna connector of the module which blocks most of the module from view.

To complete the measurements two Bluetooth Devices had to be used; one as the master and one as the slave. In the case of the Casiras, one device was programmed to automatically accept incoming connections. This device was used as the slave in the tests, as the slave only needs to be able to accept connections; it does not have to do anything with the connection once it is established. The other device was controlled by

(a) (b)

Figure 21. Cambridge Silicon Radio's (CSR) Casira Development System. (a) External View. (b) Internal View.

the Range Estimation application as previously described in Chapter 5. For one set of measurements, a BlueDolphin was used as the master, and a Casira was used as a slave. Limited success was achieved in adapting the Range Estimation application to support the BlueDolphin in master mode; however the results were unreliable enough that the Bluetooth Upper Layer Terminal Application had to be used manually to collect RSSI and Link Quality data from the BlueDolphins. The antenna ports on the two test devices were connected together by a variable attenuator as shown in Figure 22. This setup was chosen as it provides a stable and repeatable environment free from interference and multipath, while actual distance measurements are much less repeatable and less controlled.

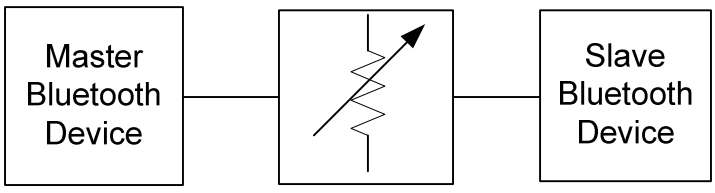


Figure 22. Test setup used for cable measurements.

Before useful RSSI versus attenuation measurements can be made, power control must be disabled on one of the devices. With power control enabled on both devices, the two devices will work together to move the reported RSSI values into the Golden Range, or as close to zero as possible. Aside from an occasional random RSSI report that is non-zero, the only time the reported RSSI values will tend to be non-zero is when the attenuation between devices is very low, or very high. In these two extreme cases the power control system will be unable to compensate, and the reported RSSI values will be negative in the case of high attenuation or positive in the case of low attenuation. Disabling power control in the slave will prevent the slave from changing its output power level. Disabling power control in the master will prevent the master from requesting a power level changed from the slave, and hence the slave's output power level will remain the same. Either approach will produce the same results that the reported RSSI values will vary solely based on the attenuation between the devices.

In the case of the Casiras, disabling power control was accomplished in the slave device by modifying the power table as described in Section 4.2.2, as this is the simplest solution. With the modified power table, the slave will not be able to adjust its power levels in response to a request from the master. The output power level of the slave will always remain the same, and the RSSI values reported by the master will only depend on the attenuation between the two devices. With no development tools available for the BlueDolphin devices, it was not possible to disable power control on these devices. Because of this the BlueDolphin could not be used as the slave and all further measurements used the BlueDolphin as a master with a modified Casira used as the slave.

Figure 23 shows two Casira devices connected through the attenuator. This setup was used in the majority of the measurements. The device on the left is connected to a computer through USB so that it may be controlled by the Range Estimation application. The device on the right is running an embedded application that automatically accepts connections from any Bluetooth Device, thus requiring no external control.

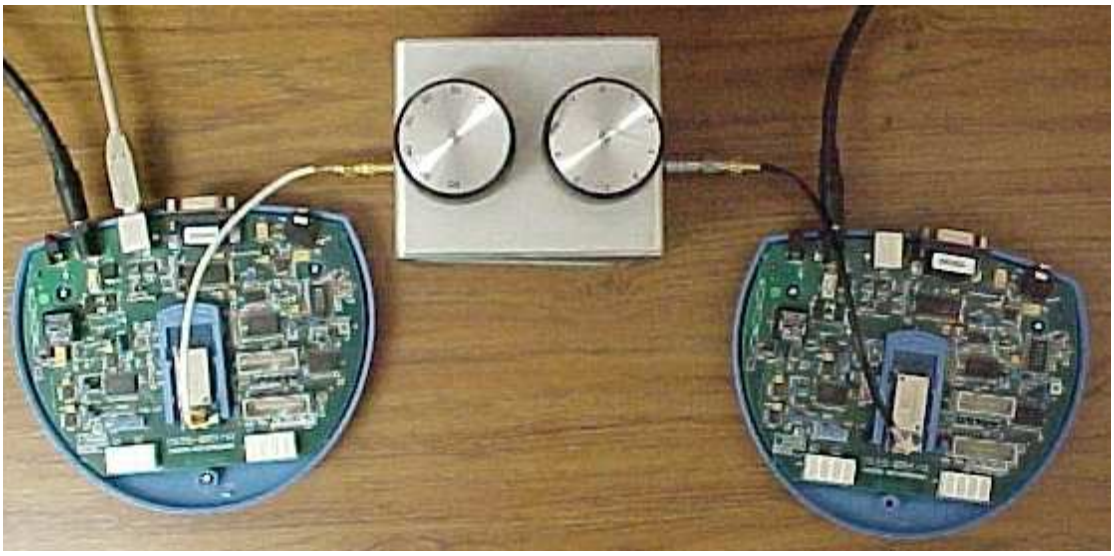


Figure 23. Two CSR Casiras setup for Cable Attenuation vs. Reported RSSI measurements.

The attenuator setup using a Casira device as the slave and a BlueDolphin device as the master is shown in Figure 24. Several reliability issues were encountered with the BlueDolphin making it difficult to establish and maintain connections without the Bluetooth Stack crashing. These issues most likely occur because the Bluetooth Upper Layer Protocol Stack was validated against CSR hardware, and was not tested for compatibility with other hardware. To obtain extra debugging information when using the BlueDolphin, both the BlueDolphin and the Casira devices were controlled manually using the Upper Layer Terminal Application. For this measurement, the BlueDolphin was controlled by one instance of the application over the RS-232 interface, and the Casira was controlled by a separate instance of the Terminal Application over the USB interface.

6.1.1. Cable Measurement Procedures

Two types of measurements were performed using the attenuator setup described in the previous section: RSSI measurements, and Link Quality measurements. The procedures used for both types of measurements are the same, and were mostly automated by the

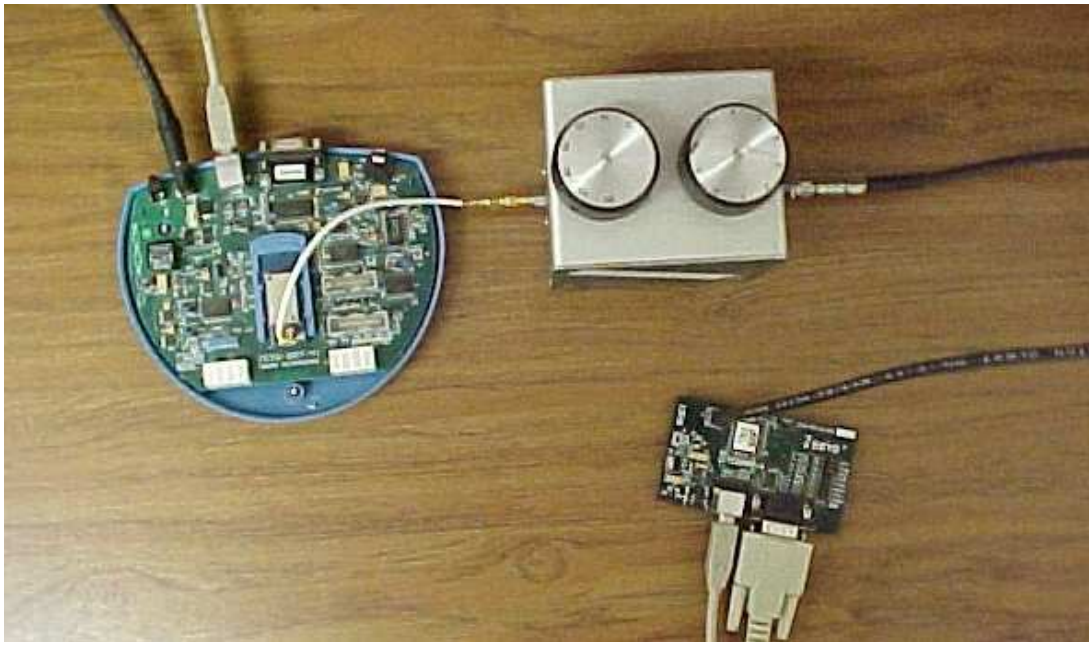


Figure 24. A CSR Casiras and a Zeevo BlueDolphin setup for Cable Attenuation vs. Reported RSSI measurements.

Range Estimation application. First the connection between the devices is established with a minimal attenuation in the cable to protect the receivers in the devices. This base attenuation was inserted between the devices as a fixed attenuator independent of the variable attenuator. For all of the data collected an initial base attenuation of 10 dB was used, however other values were experimented will to verify that the resulting data was not influenced by either attenuator. If the variable attenuator was causing unintended effects to appear in the data it was expected that the resulting data would be dependant on the setting of the variable attenuator and not the total attenuation between devices; hence changing the fixed attenuator would cause the same effects to be seen at a different setting on the variable attenuator. After verifying that the value of the fixed attenuator did not change the resulting data, 10 dB was decided on as the best value to use. This value was high enough to provide some level of protection to the receivers from being overpowered, but it was also small enough to provide a large range of attenuation values over which data could be collected. Using a larger fixed attenuator would have decreased the range over which data could be collected.

Once the connection was established, the variable attenuator was stepped 1 dB at a time, and the resulting RSSI and Link Quality values were recorded. This process was repeated until the connection was lost, and a new connection could not be established at that attenuation setting. Between each measurement set, various radio parameters were changed to test their effects on RSSI and Link Quality. These parameters, stored in the device's PSKeys, are described in sections 4.1.1, 4.1.2, and 6.1.1. As radio parameters were modified for each measurement set, the performance of the devices changed, and as a result each data set does not necessarily contain the same number of data points. Some settings caused the connections to be lost at lower attenuation levels than other settings. The Range Estimation Application automated the collection of the RSSI and Link Quality values by collecting 10 seconds worth of data and reported the mean, median and variance of the collected data as described in Chapter 5. As the attenuation between the devices increased, the number of packet errors also increased, lowering the data rate. This results in a large number of packets averaged for low attenuation values (typically just under the best case of 1,000 packets), and far fewer packets being averaged as the attenuation increased. In some cases as few as 100 packets were averaged. This is not as big a problem as it may sound because in times of high packet loss, the RSSI and Link Quality values being reported back from the master device did not vary. In these cases the received signal strength was so low that the reported RSSI values were constant at the lowest value the device was capable of reporting. While the Link Quality values were not at their minimum, they always had a variance of zero, and were assumed to be stable enough for the measurement to be considered valid even with a small number of samples.

6.1.2. Casira RSSI Measurements

The CSR BlueLab Software Development Kit (SDK) provides the ability to modify several radio parameters of the BlueCore devices. Each parameter is stored in what CSR calls a Persistent Store Key (PSKey). Each PSKey is simply a memory location, or collection of memory locations in a flash based storage element in the radio. While the PSKeys can store such information as the name of the device, they also contain configuration and calibration parameters such as the radio power table, oscillator trim

values and the Bluetooth Address of the device. Upon examining the limited documentation publicly available for the PSKeys, and trying various combinations of some of the settings, four of these parameters were found to be critical to the operation of power control in the Casiras. *PSKEY_LC_POWER_TABLE* is used to set the output power of the device in response to power control messages. In the slave devices, this was used to disable power control as previously described in Section 4.2. *PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM, PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM*, and *PSKEY_LC_RSSI_GOLDEN_RANGE* affect the operation of the Golden Range, and hence the RSSI values returned by the device. For each data set collected from the Casiras, *PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM, PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM*, and *PSKEY_LC_RSSI_GOLDEN_RANGE* were modified to determine the effect that each of these parameters had on the Golden Range. Each parameter is represented in the modules as an unsigned byte.

Table 2 lists the default values of the Persistent Store Keys that were modified before each data set was collected. *PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM* and *PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM* were varied between 1 and 12 and *PSKEY_LC_RSSI_GOLDEN_RANGE* was varied from 150 to 0 in steps of 10. While there are many valid settings outside of these ranges, setting any of the PSKeys outside of these ranges either caused the radios to stop working altogether, or caused the same RSSI value to be reported no matter what the attenuator setting was. Because of the large amount of data that would have to be collected to test all combination of values, the PSKeys were varied one at a time, while the remaining two PSKeys were kept at their default values. From the collected data the best results were chosen as the new starting point and the PSKeys were varied again.

Persistent Store Key	Default Value
PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM	12
PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM	1
PSKEY_LC_RSSI_GOLDEN_RANGE	80

Table 2. Default Persistent Store Key values in the Casira Development Kits.

6.1.3. BlueDolphin RSSI Measurements

Because CWT did not have a development kit for the BlueDolphin available, much less data was collected using it than was collected from the Casiras. Without a development kit, the BlueDolphins had to be used as is. No adjustments to the radios could be made. This means that the power control mechanism in the BlueDolphins could not be disabled, making them unsuitable for use as slaves in a RSSI test. This also means that there are no parameters to tweak as there are in the Casiras, and with nothing to vary between data sets, only one data set was collected. The BlueDolphins further presented problems in that their connections were not stable, and they often caused the Bluetooth Stack to crash. It was very difficult to keep the BlueDolphins connected for more than a few seconds, and because of this, it was not possible to automate the data collection process. Instead the Bluetooth Upper Layer Terminal Application was used to manually request RSSI reports one at a time, and these values were recorded. This method was successful as a failed connection could be manually restarted and data collection could be resumed manually. Because the process was not automated, only 10 RSSI samples were collected to be averaged for each data point, and in all cases the BlueDolphin reported the same RSSI for each of the ten samples. As in the case of the Casira modules, data collection was started with minimal attenuation in the cable, and the attenuation was increased in steps of 1 dB for each data point collected. Since connection failures were common, even with low attenuation in the cable, data collection continued until a connection could not be sustained long enough to retrieve a single RSSI value, even after repeated attempts.

6.1.4. Casira Data

While 49 sets of data were collected using two Casira kits and the attenuator, only the most interesting plots are presented in this section. All of the data are shown in Appendix A. Figure 25 shows the mean RSSI reported by the Range Estimation Application vs. the total attenuation in the cable for the default settings of the Persistent Store Keys. It is important to note that the RSSI values reported by the Range Estimation Application in these test are not the same as the RSSI values reported by the radio. The values shown in the figure have been adjusted to take into account the Golden Range as described in Chapter 5. The effects of the Golden Range in Figure 25 start when the total cable attenuation reaches 21 dB and end when the attenuation reaches 37 dB, resulting in a width of 16 dB. While this range is slightly smaller than optimum, it is within the specifications. The effect of the adjustment to the RSSI values by the Range Estimation Application can be seen on either side of the Golden Range as the large jump in RSSI values from 0 to 12 at the top of the Golden Range and from 0 to -20 at the bottom of the Golden Range. Without the adjustments the discontinuities would be much smaller: only from 0 to 2 at the top of the Golden Range and 0 to -10 at the bottom of the Golden Range; however the adjustment is included to provide a linear scale for RSSI. With the adjustment, a change of 1 in the RSSI data would ideally equal a change of 1 dB in attenuation. The goal is to properly adjust the PSKey values such that the plot of adjusted RSSI vs. attenuation is linear, and the width of the Golden Range has been reduced as much as possible.

Figure 26 shows the first combination of PSKey settings that were tested aside from the default values. This plot illustrates why more measurements are not needed. Outside of the range of settings presented, the result was often that all of the reported RSSI values were -20, the lowest RSSI value that the BlueCore02 chipset will report. In some extreme cases, such as setting the value of *PSKEY_LC_RSSI_GOLDEN_RANGE* above 150, the radios would either stop working completely, or would work intermittently, preventing a connection from being established or maintained long enough to collect enough data.

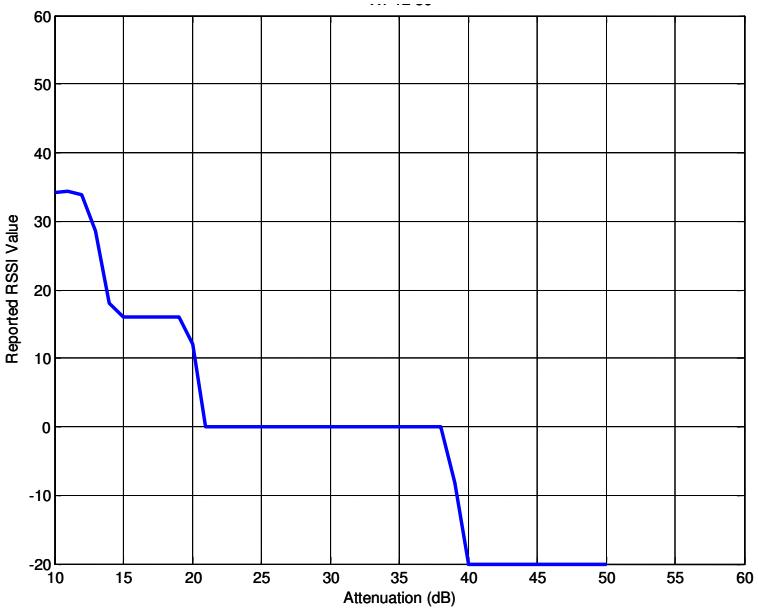


Figure 25. Mean Reported RSSI vs. Cable Attenuation wih the default PSKey settings.

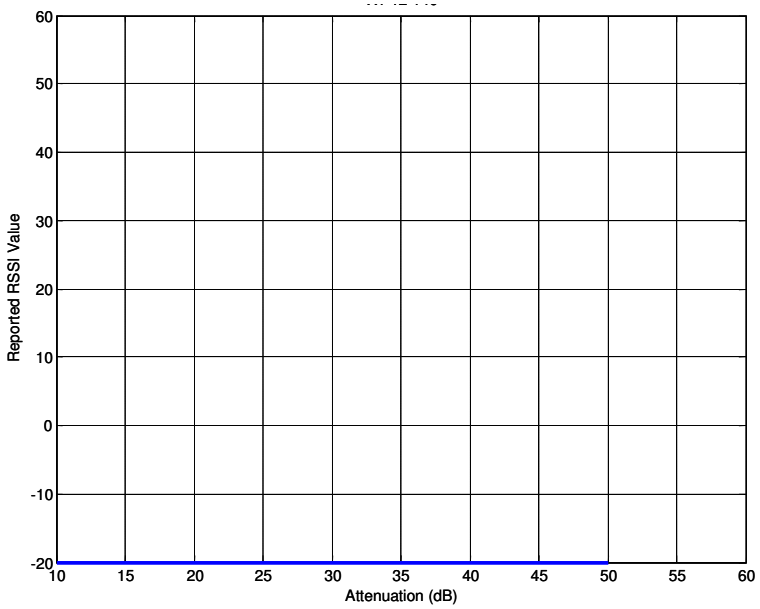


Figure 26. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=150.

Figure 27 is representative of almost all of the issues that are seen in all of the other data sets. While most of the other data sets do not contain all of the issues present here, all of the data sets contain at least one of the issues. The largest problem with the combination of settings used in Figure 27 is that the resulting curve is not monotonic. This is an issue that is seen in many of the other data sets as well. A second issue that arises is that the reported RSSI values only take on a few values, creating plateaus in the plot. This is expected in the case of the Golden Range, but was not expected outside of the Golden Range. In Figure 27, there are five other plateaus which are more than what occurs in the majority of the data sets; however all of the data sets present the same problem. Finally almost no useful information is reported when the received signal drops below the Golden Range. As received signal strength decreases, the reported RSSI values will change from 0 to -20 with only a few intermediate steps if any. In Figure 27, it can be seen that there is one data point between 0 and -20 and this was caused by the averaging of the reported RSSI values consisting of only 0 and -20. All of these effects will cause a problem in a range estimation application because of the ambiguity that they introduce into the range estimate. For example, an application would not be able to tell if the cable attenuation was 22 dB, 19 dB, or 16 dB from Figure 27.

The first set of tests tried to determine the effect that modifications to *PSKEY_LC_RSSI_GOLDEN_RANGE* had on the operation of the radio and the reported RSSI values. The test consisted of varying *PSKEY_LC_RSSI_GOLDEN_RANGE* from 150 to 0 in steps of 10 while keeping the other two PSKeys constant, and recording the mean of the reported RSSI value versus cable attenuation for each configuration. From this data, the default PSKey settings were determined to be the best. Compared to the other settings tested, the default configuration is monotonic, and it also provides more useful (smaller plateaus yielding fewer ambiguous points) data points, and the connection tends to be more stable.

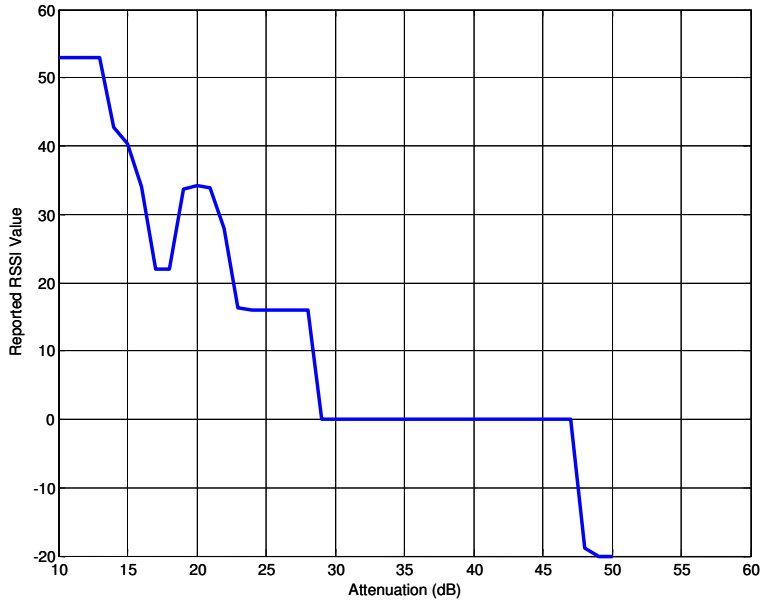


Figure 27. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=50.

The next test was started with the default PSKey settings again, and this time *PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM* was varied, starting with the default setting of 1 and incremented in steps of 1 up to 12. Figure 28 shows the plot of RSSI versus attention with *PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM* equal to 12. Here it can be seen that the Golden Range is almost non-existent; however the range over which an RSSI of -20 is reported is rapidly increasing; setting *PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM* to13 will cause all reported RSSI values to be equal to -20, and increasing it beyond that will cause the radio to stop working. Next the default PSKey settings were restored and *PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM* was varied from its default of 12, down to 1. From the 24 sets of data generated by varying these two PSKeys, the best settings, the results of which are shown in Figure 29, were selected and used as the starting point for the next set of tests. This configuration shown in Figure 29 shows a lot of improvement over the default settings. The width of the Golden Range has been reduced to only 5 dB, and while the curve is not monotonic, there is a 10 dB wide range over which useful

information can be extracted. Although RSSI in this range is not linear with respect to attenuation between devices, this could be corrected for in post processing.

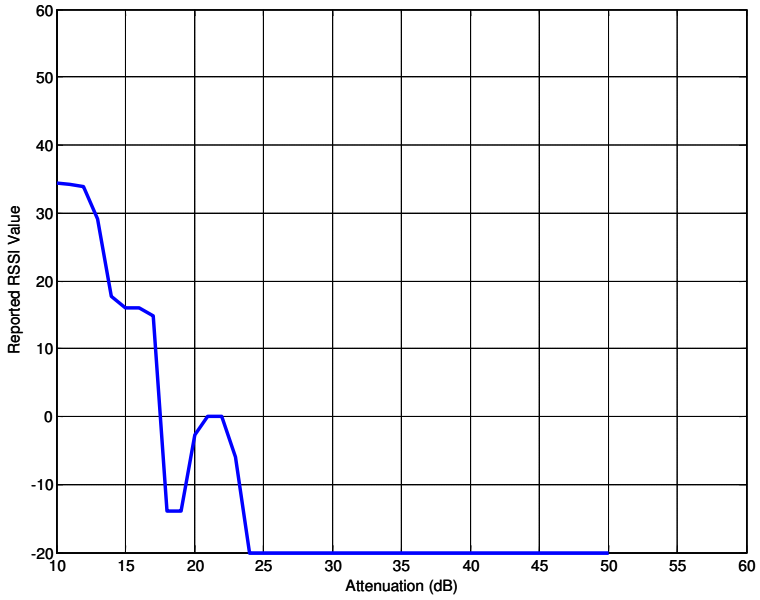


Figure 28. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=12 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

Finally starting with the PSKey settings from Figure 29 *PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM* was varied from 1 to 12, as modifying this PSKey seemed to cause some improvement in the earlier tests. The final result is shown in Figure 30, with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=3 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4, and PSKEY_LC_RSSI_GOLDEN_RANGE=80. There are several minor improvements over the settings shown in Figure 29, the width of the Golden Range has further been reduced, and the dip in the curve just below 20 dB of attenuation is not as deep, although it is wider. These settings, while still far from ideal provide the widest useable range of any of the settings tested.

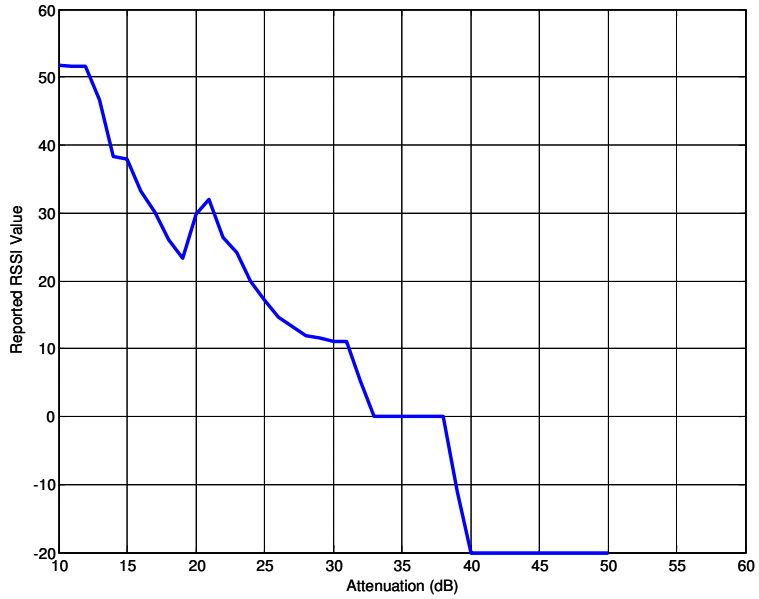


Figure 29. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

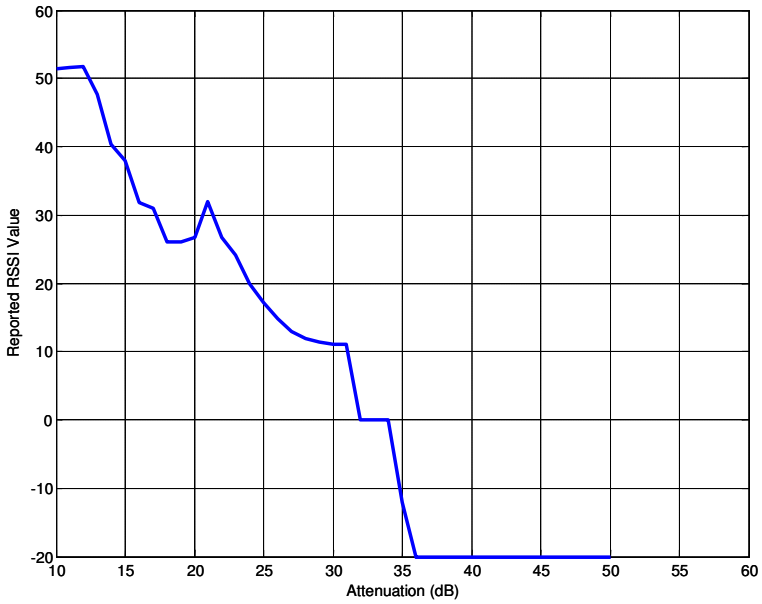


Figure 30. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=3 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

6.1.5. BlueDolphin RSSI Data

Without being able to disable power control in the BlueDolphin, it could not be used as the slave device in the measurements. If used as the slave device with power control enabled, the BlueDolphin would adjust its output power level in response to power control requests from the master device, in this case one of the Casiras. With the exception of a few random packets, all of the packets received by the master would fall within the Golden Range of the master, and the reported RSSI values would be zero. This only leaves the option of using the BlueDolphin as the master device, and using a Casira with power control disabled as a slave device.

Since there were no radio parameters of the BlueDolphin that could be modified only one data set was collected using the BlueDolphin as the master and a Casira with power control disabled as the slave. Maintaining a connection between the BlueDolphin and the Casira was difficult with problems caused by PC side stack crashes, and frequently dropped connections. Because of these problems, the process of collecting RSSI data from the BlueDolphin could not be automated. Each data point was collected manually, and as a result fewer RSSI values were averaged for each data point than were averaged in the previous Casira measurements. Only 5 RSSI values were averaged to create each data point; often the connection would fail and have to be reestablished in order to collect the 5 RSSI values for one data point. Figure 31 shows the data collected while using the BlueDolphin as the master device. The data shown are the average of the raw RSSI values collected from the BlueDolphin; these values were not corrected for the Golden Range as was done with the RSSI data collected from the Casiras. While the width of the Golden Range meets the specifications, almost no other portion of the graph complies.

A connection could not be established or maintained with less than 20 dB of attenuation or more than 60 dB of attenuation in the cable. As the attenuation between the devices was increased from the starting value of 20 dB, the reported RSSI values decreased as would be expected. However, after getting to what should be the bottom of the Golden Range the reported RSSI values started to increase again, instead of decreasing as

expected. The reported RSSI values were never negative at any attenuator settings. This response will not work in a range estimation context, and it raises some questions about the effectiveness of power control on the BlueDolphin devices.

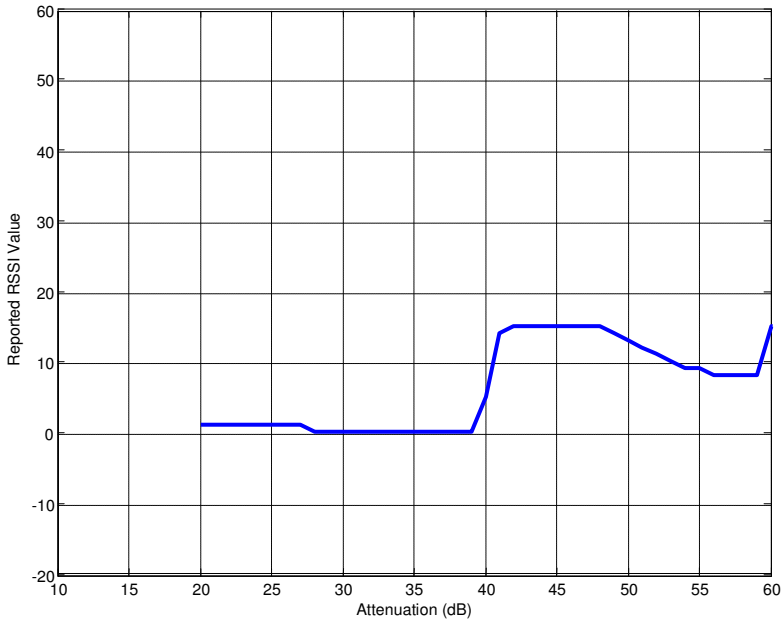


Figure 31. Cable Attenuation vs. Reported RSSI using a BlueDolphin as the master.

Without ever generating a negative RSSI value, the BlueDolphin will never request a power increase. From Figure 31, once the received signal strength drops below the Golden Range and the BlueDolphin will constantly be generating decrease power request messages, contrary to what would be expected. If the other device is able to comply with all of the requests, this will cause the received signal strength to eventually drop to an unusable level, and the connection will be lost. More likely is that the other device will not be able to decrease it output power indefinitely, however this will still cause undesirable results manifested as a higher than expected bit error rate.

6.1.6. Casira Link Quality Measurements

Link quality measurements were performed with two different configurations of the PSKeys using two Casira kits. The setup for the Link Quality measurements is the same as for the RSSI measurements. The Bluetooth Specification states that the calculation and interpretation of Link Quality is completely up to the hardware manufacturer. One interpretation of this specification is that Link Quality has no meaning and never changes. Even if Link Quality has a significant meaning in the Casiras, it may or may no be dependant on any of the radio parameters that can be changed. For this reason the Link Quality measurements were initially performed with the default PSKey settings and with the PSKey settings that produced the best RSSI versus attenuation curve. Figure 32 shows the results for the default PSKey settings and Figure 33 shows the results for the PSKeys that produced the best RSSI curve.

The Link Quality graphs provide very little useful information, and modifying the PSKeys has little effect on the results. The Casiras reported a Link Quality of 255 until there was noticeable packet loss, and then the reported value started to decrease slightly. Because of the lack of information contained in the Link Quality measurements and the lack of ability to make a modification to the radio that would change the results, no further Link Quality measurements were conducted.

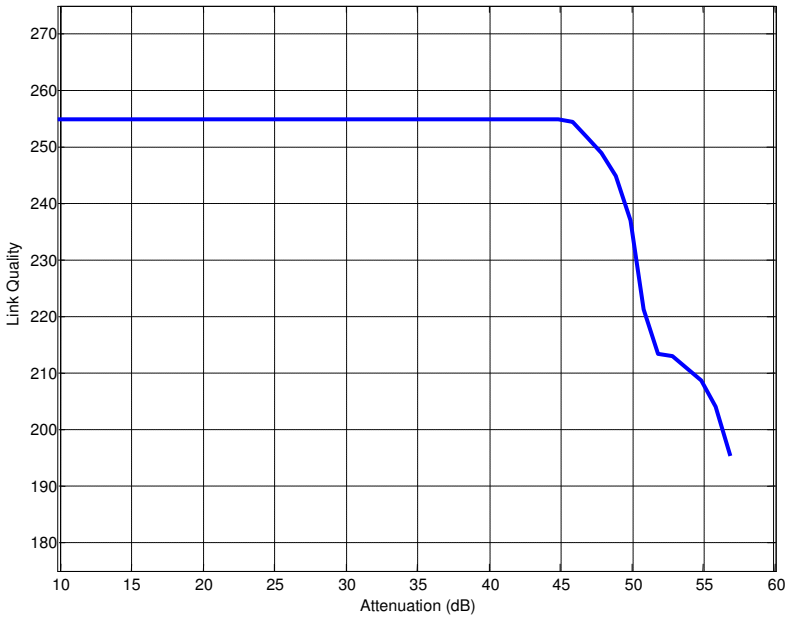


Figure 32. Link Quality vs. Attenuation for the default PSKey settings.

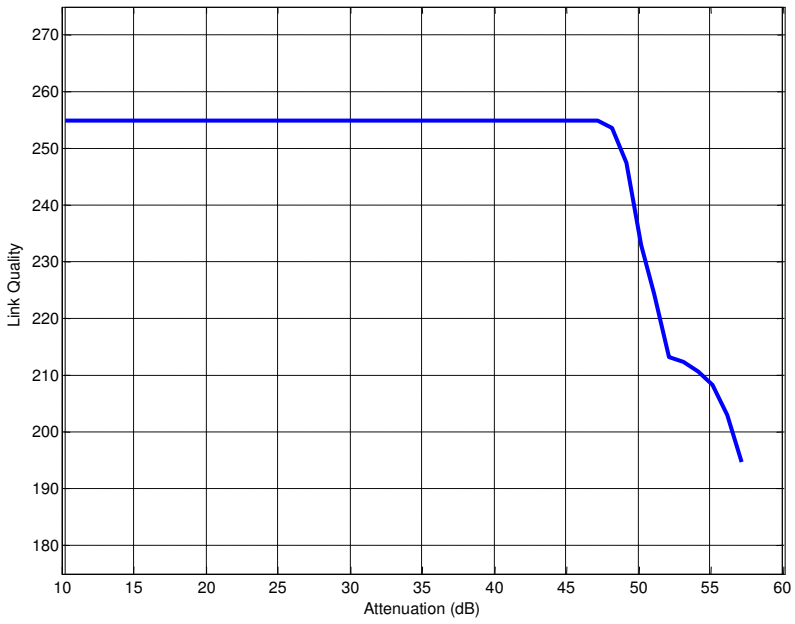


Figure 33. Link Quality vs. Attenuation with optimal PSKey settings from RSSI measurements.

7. Distance Measurements

Using the best PSKey settings as determined from the cable RSSI measurements, a series of indoor and outdoor distance measurements were performed to try and correlate a reported RSSI value with a distance. For each set of measurements, two devices were positioned three feet apart, and RSSI statistics were collected in the same process that was used with the cable measurements. The master device was then moved 1 foot farther away from the slave device, and the process was repeated until the connection failed and could not be reestablished. For these measurements a Uniwill module, shown in Figure 34, incorporating the CSR BlueCore02 chipset was used instead of the Casira Kits. The Uniwill modules contain the same chipset as the Casiras, and were used here for their portability. In contrast to the Casiras, the Uniwill modules can be battery powered, or run off of the power from a USB port of a laptop. The Casiras do not have this capability, and must be plugged in making them less than ideal for situations in which they must be moved often.

The Uniwill modules are mounted on a PC board designed at CWT. The board contains all of the necessary external hardware to operate the module, including a ceramic antenna which can be seen in the upper left corner of Figure 34. The PC board includes a voltage regulator allowing the module to be powered with anywhere from 3.3 to 12 volts, allowing for a wide range of power supply options. Unlike the Casiras there are no test points, or transport interface connectors other than the USB connector, making the board small and uncluttered, which is ideal for this type of measurement. The silver shield seen on the left side of the board contains the Bluetooth module and is the same size as the module in the Casiras.

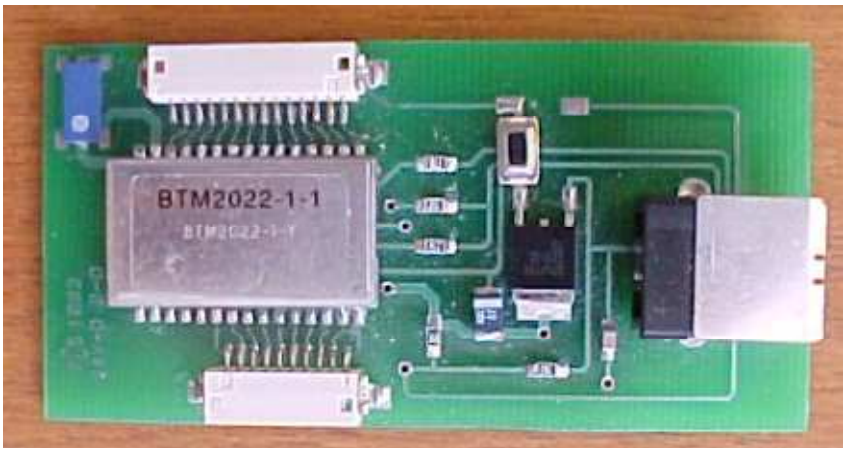


Figure 34. Uniwill modules used in distance measurements,

7.1. Indoor Distance Measurements

The indoor distance measurements were performed in the hallway of CWT's Modular Building on the Virginia Tech Blacksburg campus. The building is of standard wood framing and drywall construction typical of residential and small office buildings. The section of the hall used for the measurements is just over 3.5 feet wide and 35 feet long. Figure 35 shows a portion of the blueprints from the building, including the hallway. To determine how reproducible the results of the measurements were, four sets of indoor measurements were conducted. The procedures for each measurement set are identical, and the second set of measurements was performed immediately after the first set. The final two sets of measurements were performed the following day, and again the fourth set of measurements was conducted immediately following the third set.

In all of the measurement sets, the slave device remained stationary at one end of the hall and the master was moved just prior to recording each data point. The location of the slave device is indicated in Figure 35 along with the start and stop locations as well as the path along which the master device was moved. In all four sets of measurements, the master device started 3 feet away from the slave, and was moved backwards 1 foot for each data point until it was 28 feet away from the slave. In this situation, the maximum

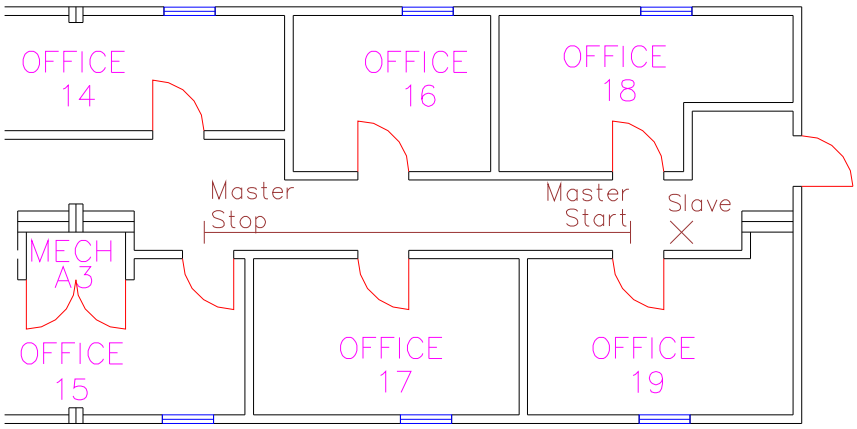


Figure 35. Portion of Modular Building used for indoor measurements

distance over which measurements could be performed was limited not by the performance of the radios, but by the length of the hallway. Figure 36 and Figure 37 are pictures of the measurement setup from the slave side and from the master side respectively. Both pictures were taken approximately halfway through a measurement set, and the distance between devices is roughly 15 feet. In Figure 36, the slave device can been sitting on the far edge of the closer of the two carts. The master device cannot be seen because of lighting conditions in the picture, but it is placed on the near edge of the far cart similarly to the positioning of the slave device. In Figure 37, the master device is directly behind the laptop at the edge of the cart, and is not visible. In all of the distance measurements made, the modules were at the edge of the carts. This put the antennas in a vertical polarization 2.67 feet above the ground.

Figure 38 and Figure 39 show the data collected from the first two sets of indoor measurements as well as exponential and linear regressions on the data. From the analysis of Section 3.1.3.1, an exponential regression should provide the best fit if the reported RSSI values are linear with respect to actual received power. The results of the RSSI cable measurements show that this is not the case, and, in the useful range, the reported RSSI values are closer to exponential with respect to received power. The results from the remaining two measurement sets are presented in Appendix B.

Figure 36. Measurement setup in the Modular Building as seen from the slave side The slave device is sitting at the far edge of the near cart.

Figure 37. Measurement setup in the Modular Building as seen from the master device side.

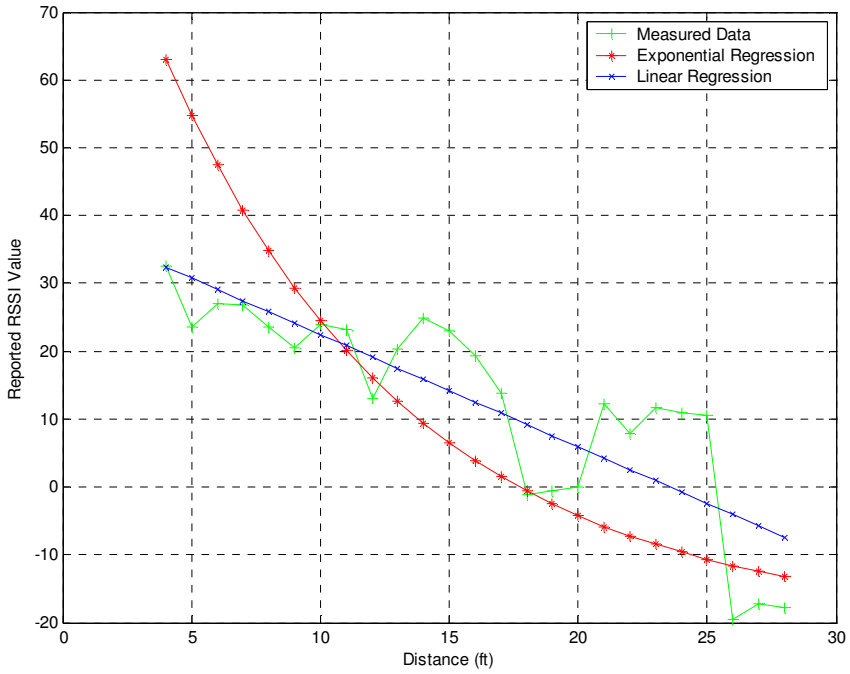


Figure 38. Indoor measurement set #1.

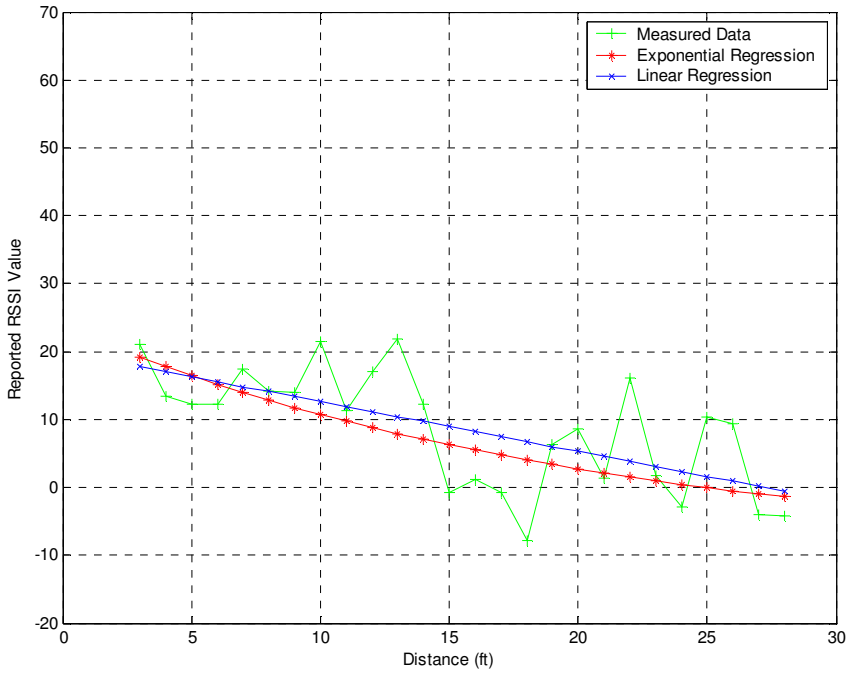


Figure 39. Indoor measurement set #2.

Although there is a definite trend in the data for the RSSI values to decrease as range increases, it is difficult to determine if the trend is linear or exponential due to the large amount of random variation in the data. Although the trend should be exponential, both exponential and linear regressions on the data are presented because of apparent randomness of the data. Eq. 13 and Eq. 14 define the linear and exponential regressions respectively.

$$range = m \cdot RSSI + b \qquad \textbf{Eq. 13}$$

$$range = b \cdot m^{RSSI} + c \qquad \textbf{Eq. 14}$$

Table 3 lists the regression statistics for each of the indoor measurements sets, and Table 4 lists the regressions statistics for the outdoor measurements sets. The Statistic R indicates how well a given regression matches the data that it is trying to approximate. R ranges from 0 to 1, and if R is equal to 1, the regression is an exact fit to the data, and if R is equal to 0 then there is no correlation between the regression and the data. For a regression to be a good fit, and be a useful prediction, R should be greater than 0.9 or 0.95 depending on the application, any regression with a smaller value of R will probably not be very useful. The largest value of R from any of the data sets is 0.698 from the linear regression of data set number 2. From Figure 39, it can be seen that while this regression is not bad at estimating the overall trend, it does a poor job of estimating any one data point because of the random variation in the data. It is interesting to note that in all of the indoor measurement sets, the linear regression does a better job of estimating the data than the exponential regression. This is the opposite of what was expected and is probably due to the poor correlation between actual received signal strength and the reported RSSI values, as well as other factors in the Modular Building such as multipath and interference from other devices. It was expected that the dominate loss mechanism would be free space loss, however this is not what is seen in the data. Because of the position and orientation of the antennas, it is likely that there is a large multipath component present from ground bounce.

A larger problem than the lack of a good regression fit is the lack of repeatability in the measurements. There is a large difference between each of the data sets. Each measurement took approximately 15 minutes to complete, and the measurements of Figure 38 and Figure 39 were completed back to back, with the entire process taking about 40 minutes. Even if a regression fit was not possible from the data, a lookup table could be used if the measurements were repeatable. However, with the lack of repeatability over even such a short period of time, it would be impossible to make use of this data for a range estimation application.

Measurement Set	m	b	R
1	-1.660	39.065	0.698
2	-0.922	20.690	0.420
3	-0.734	19.965	0.415
4	-0.655	20.254	0.321

Table 3. Linear regression statistics from the indoor measurements.

Measurement Set	m	b	c	R
1	0.901	125.971	-20	0.432
2	0.959	44.951	-15	0.360
3	0.972	41.820	-10	0.388
4	0.978	40.193	-15	0.219

Table 4. Exponential regression statistics from the indoor measurements.

7.2. Outdoor Distance Measurements

The outdoor distance measurements were done in the same fashion as the indoor measurements. The primary difference being that longer distances were achievable as available space in which to conduct the outdoor measurements was not a restricting factor. Each set of measurements was conducted up to the maximum range at which a connection could be sustained, with this value varying substantially between

measurements. Coincidentally, the first measurement set had the shortest maximum range at only 17 feet, while the last measurement set had the longest range at 40 feet. Figure 40 shows a diagram of the location of the outdoor measurements including the location of the slave device, the path over which the master device was moved, and the surrounding buildings. Although the measurement sets ended at different distances, only the stopping location of set number 4, the longest set, is shown. The location chosen for the outdoor measurements was just outside the opposite end of the Modular Building used for the indoor measurements. This location was picked because of its easy access, and open space. Without the walls and office furniture of the Modular Building surrounding this location it was hoped that many of the random effects seen in the indoor measurements would not be present. The measurements were made at one edge of a 21 foot wide sidewalk, bordered by a small lawn. This placed the Bluetooth Devices over a consistent surface and in the middle of the open space for the majority of the measurements. The edge of the sidewalk is not depicted in Figure 40.

Figure 41 is a picture taken from the middle of the sidewalk showing the slave and the master in its starting position. The master device is on the right side of the left cart, and the slave device is on the left side of the right cart. Because of the size of the devices, it is difficult to see them in the picture. From its position in the picture, the master device was moved to the left while the slave remained stationary throughout the measurements. From the picture of Figure 41, the slave appears to be much closer to the modular building than indicated in Figure 40, this is because the stairs and ramp seen in the background of the picture are not depicted on the diagram, and neither is the small sidewalk seen on the upper left side of the picture.

The timing of the outdoor measurements was the same as for the indoor measurements. Two sets of measurements were taken on the first day, the second one immediately after the first one. At the same time on the second day, the third and fourth sets of measurements were taken, again one immediately after the other. The results of the first set of outdoor measurements are shown in Figure 42, while the results of the last set of

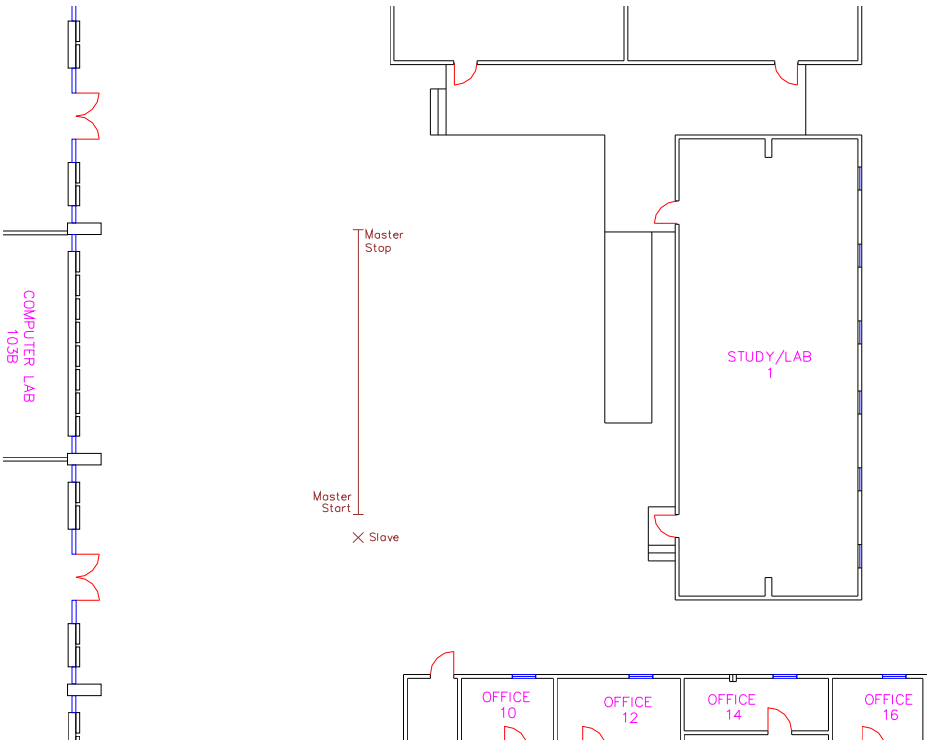


Figure 40. Location of outdoor distance measurements.

Figure 41. Outdoor distance measurements showing both master and slave devices.

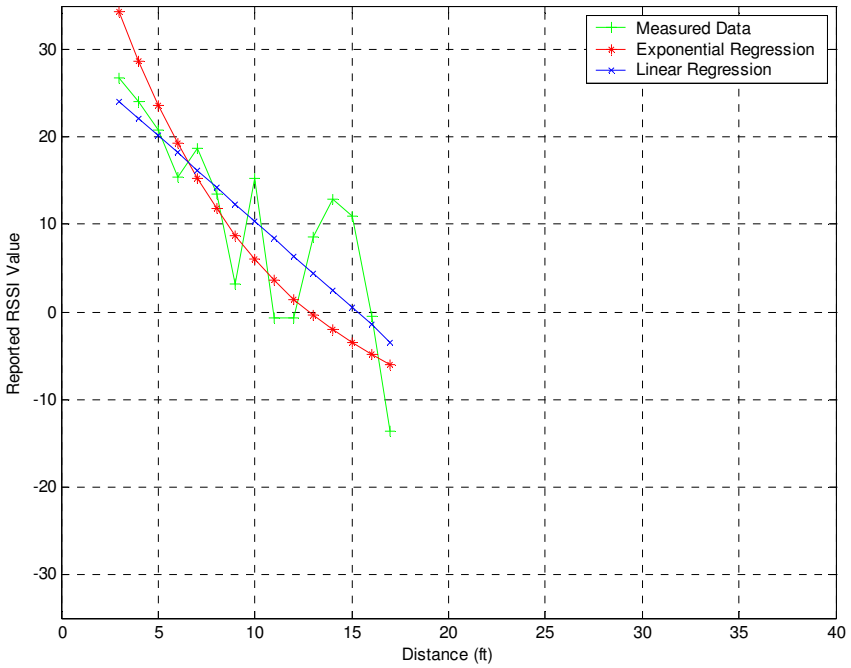


Figure 42. Outdoor measurement set #1.

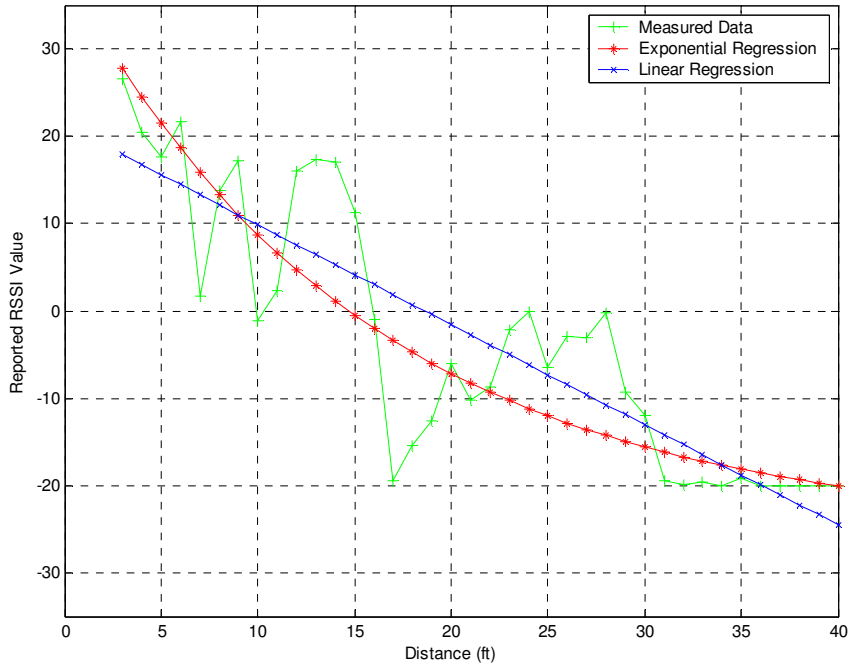


Figure 43. Outdoor measurement set #4.

outdoor measurements are shown in Figure 43. The data from the second and third sets of measurements are presented in Appendix B. The first and fourth set of measurements were chosen for presentation here as they best illustrate the differences between the indoor and outdoor measurements while still showing the issues that remain even in open areas. It is surprising that even in a relatively benign environment there is a fair amount of randomness associated with the data. It is probably due to this random variation in the data that the linear regression is a better fit than the exponential regression, even though neither is a particularly good fit. The regression fit statistics for the linear and exponential regressions are listed in Table 5 and Table 6 respectively.

Measurement Set	m	b	R
1	-1.965	29.969	0.642
2	-1.674	21.151	0.767
3	-1.862	28.789	0.693
4	-1.145	21.320	0.734

Table 5. Linear regression statistics from outdoor measurements.

Measurement Set	m	b	c	R
1	0.922	62.067	-15	0.523
2	0.910	51.425	-20	0.605
3	0.891	74.814	-20	0.281
4	0.938	63.982	-25	0.729

Table 6. Linear regression statistics from outdoor measurements.

The data shown in Figure 43 are the worst behaved of the outdoor data collected, with a change of over 35 in RSSI over a distance of only 3 feet. While this is still better than the worst of the indoor data, when compared to Figure 42, it shows the same repeatability problems that were seen in the indoor data. As expected, the regressions are a better fit to the outdoor data then the indoor data. However with the amount of randomness in the data the regressions statistics are almost meaningless. Although the R statistic predicts a decent fit in some cases, the resulting regression will not produce a good range estimate.

In all of the outdoor data sets the linear regression still performs better than the exponential regression, however in the last data set the exponential regression is almost as good as the linear regression.

7.3. Range Estimation Accuracy

To determine the amount of error that would be present in a range estimate, three more outdoor measurement sets were performed. This time the Range Estimation Application was used to calculate a range estimate based on previously measured data, and the estimated range was compared against actual range. The first outdoor measurement set was used as a basis for estimating range as an exponential regression. An exponential regression produces a good fit on the first half of that data; that is until the received signal strength starts to enter the Golden Range. The regression statistics on the first half of this data are listed in Table 7.

The Range Estimation Application was modified to create a range estimate from each data point using Eq. 14 and the statistics from Table 7. As described in Chapter 5 a moving average of the last 50 RSSI values was used as the input to the estimation algorithm, with a new input generated for every RSSI report. From the R statistic in Table 7, this is the best fit of any of the regressions performed so far. Using this regression to estimate distance based on mean RSSI, the next three measurement sets were conducted in the same manner as the previous outdoor set, except that estimated range between device was recorded instead of RSSI. Again, the data collected in these measurements were limited by the maximum range at which a connection could be maintained and not physical constraints. Because of this, the number of data points in each measurement set varies.

m	b	c	R
.934	52.80	25	.845

Table 7. Regression Constants use to estimate range.

The result of the first set of range estimation measurements is shown in Figure 44 and Figure 45. Figure 44 shows the range estimated by the Range Estimation Application versus the actual range using the exponential estimate defined by the statistics in Table 7. Figure 45 shows the error in each estimated range, defined by:

$$Error = Estimated\ Range - Actual\ Range \qquad \textbf{Eq. 15}$$

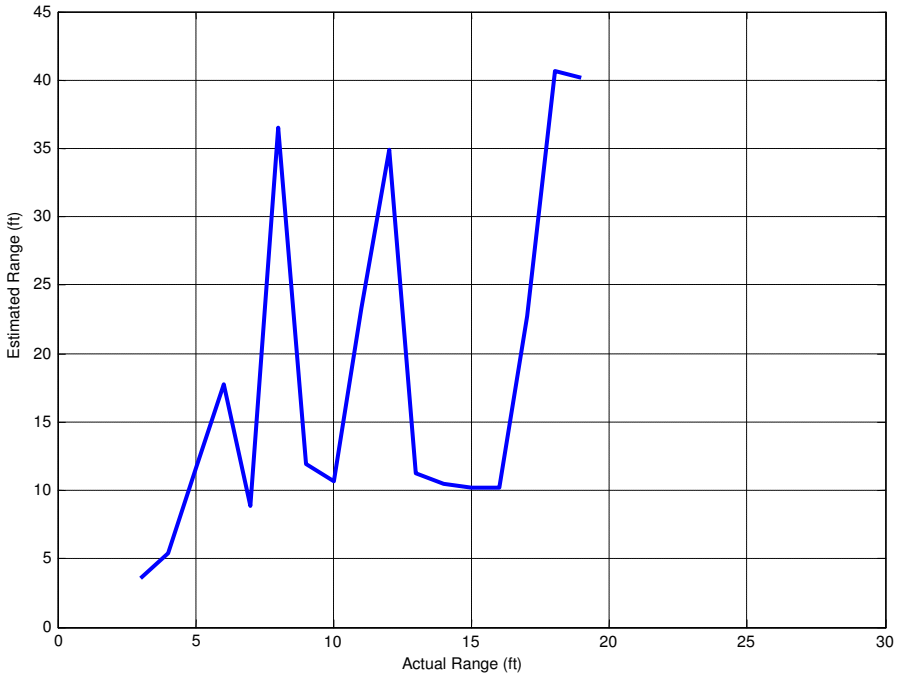


Figure 44. Outdoor Range Estimation set #1.

The other two range estimation measurements are not presented here as they are similar to the first set. The data collected in these measurement sets are presented in Appendix B. The data in Figure 44 show the same random behavior seen in the previous measurements, and while some of the data points are fairly accurate (just under half of the points contain less than 5 feet of error) there are a few random points that contain a lot of error. The worst case error is at 8 feet of actual range in Figure 44, where the estimated range is 36 feet, a 350 per cent error. The remaining two data sets show the

same behavior, with as many as 68 per cent of the data points exhibiting a low amount of total error, the remaining random data points however tend to have high error on the order of 10 to 30 feet, with no way of distinguishing an accurate data point from one with a high amount of error. It is interesting to note that although this estimate was intended to be good out to approximately 9 feet, there is still a significant amount of error prior to 9 feet, and the worst case error occurs just before 9 feet.

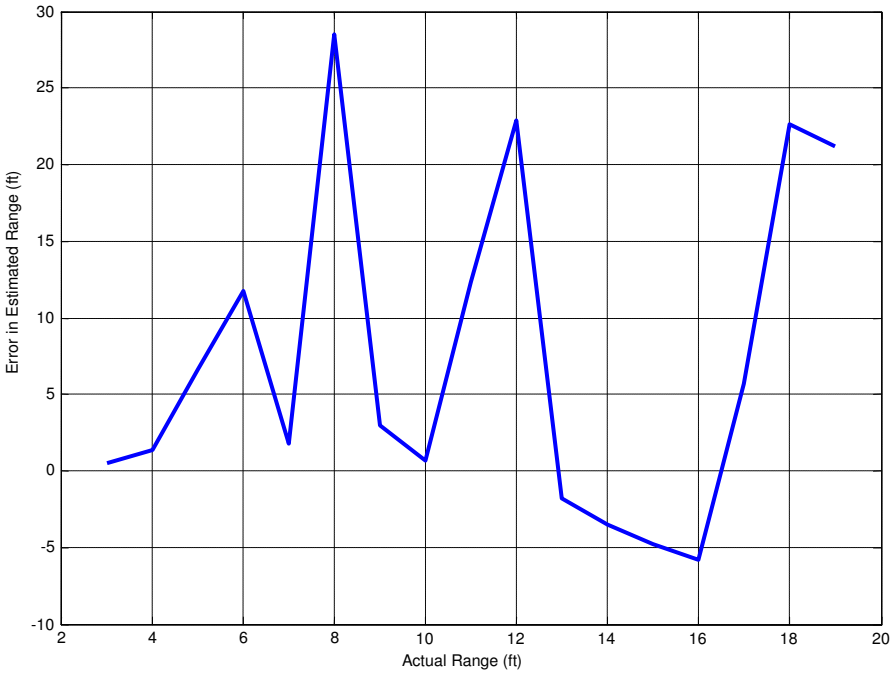


Figure 45. Error in Outdoor Range Estimation set #1.

7.4. Error Analysis

Range error may be translated into area error (the area in which a device is likely to exist around an estimated location) graphically by calculating the intersection of the rings containing the range error. Figure 46 illustrates this concept for 30 feet of estimated range with 5 feet of assumed error. Figure 46 (a) shows the range measurements including error made from two stations to a common location at the intersection of the rings. Figure 46 (b) shows the intersection of the two rings representing the total area

error in the measurement. The two gray rings represent 30 ± 2.5 feet from two locations capable of measuring range to a third location, which is the best case error from the range estimation measurements. This is the same as described in Section 2.1.3 with the addition of an uncertainty of ± 2.5 feet surrounding the measured range. The case shown in Figure 46 (a) represents the best case error in a two-dimensional system. When the rings of estimated range intersect at right angles, the error will be minimal, and when the rings completely overlap the error will be its maximum. The amount of area error is then related to the range and geometry between the range measuring devices and the target device as well as the range error in each measurement. The total area error of the example shown in Figure 46 is 25 ft^2; in the best case as shown, the area error may be computed by assuming that area is a square with the total uncertainty comprising each side; that is the area error is the square of the range error. [15]

The worst case area error will be when the two range measuring stations are collocated, or their error circles completely overlap. This was the case in several of the range estimation measurements when the range error was much greater than the actual range. In this case the area error will simply be the same as the area of the smaller error circle. This use of two range measurements in a situation such as this provides no improvement in the estimation of the location of the target. In the worst case data from the range estimation application, the error is so large, 3.5 times the actual range, that only information that can be derived from the measurement is that the target device is within the visible range of the measuring device, which was already known when the devices were able to establish a connection.

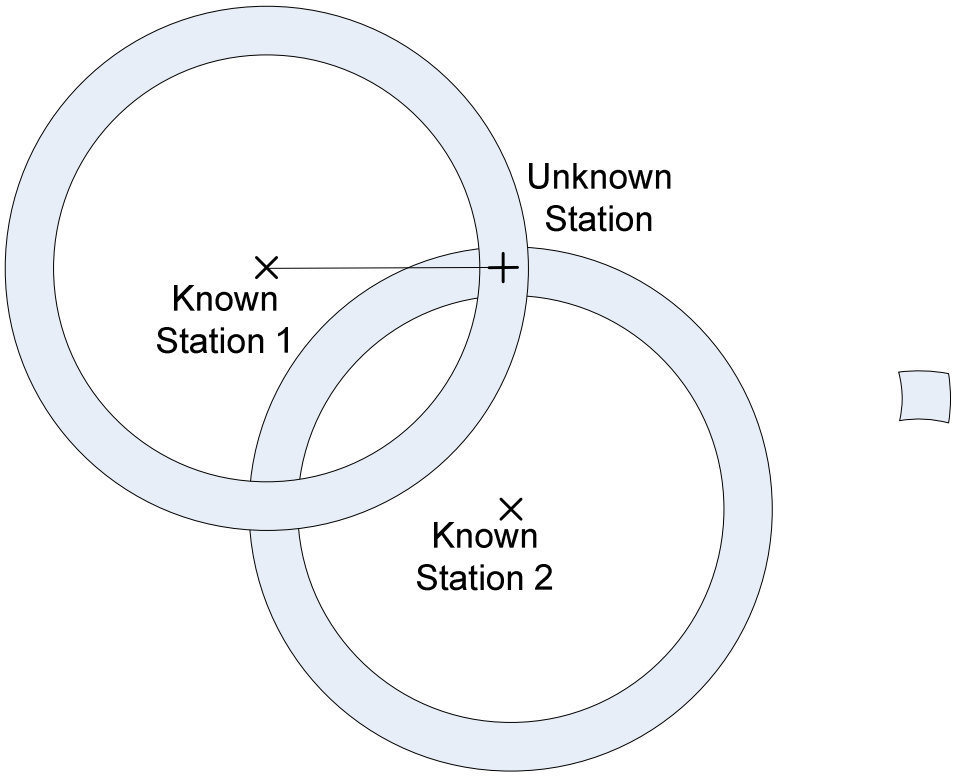


Figure 46. Graphical representation of area error from range error at 30 feet with 5 feet of uncertainty. (a) Uncertainty in range measurements. (b) Area error calculated from (a).

8. Conclusions

Although it may be possible to determine the range between two Bluetooth Devices using a RSSI based technique, it is not possible to do so using BlueCore02 or Zeevo based devices. Using RSSI and The Friis Transmission Formula to determine the range between two of these devices ultimately results in a solution that is no better than what can be achieved by simply noting whether two devices are close enough to establish a connection or not.

Devices from manufacturers other than CSR and Zeevo may perform much better in accomplishing this goal; however this is not likely the case as the primary goals of Bluetooth are cheap and simple devices that could enjoy widespread use. In meeting these goals manufactures have often developed devices that meet the bare minimum specifications. In any case, it is unfortunate that CSR devices do not perform better because the BlueCore chipset is one of the most popular chipsets in use today.

The Bluetooth Power Control Message format reserves one byte in each request for future use. Eventually this byte will be used to request specific power step sizes in the LMP_incr_power_req and LMP_decr_power_req messages. It was speculated that this would be implemented in the 2.0 revision of the Bluetooth Specification, but the current release of the 2.0 Specification still lists these bytes as reserved. Along with the ability to request specific power step sizes will come the requirement of being better able to measure the actual received power level, and it is possible that this is being implemented in 2.0 compliant devices in anticipation of it being required. Devices are becoming available that are based off of the Bluetooth 2.0 Specification. Unfortunately none of these devices were tested as modules are not yet available, and the devices are only available in BGA packages, which could not be handled. CSR is currently working on firmware revision 19.x that in the near future may provide more accurate RSSI reports, however it is reported that the hardware is unable to make measurements that are much more accurate that what can be obtained from the current firmware.

With the newer revisions of both hardware and firmware being released, it may be possible in the near future to implement the techniques described here to determine the range between Bluetooth devices.

9. Summary

This thesis looks at the possible methods of using off the shelve Bluetooth Devices to implement a position location service with accuracy greater than the service area of a single device. Because of Bluetooth's design, there are not many ways in which this can be accomplished, and the most promising of these is through the use of Bluetooth's Receive Signal Strength Indicator to estimate range between devices. Unfortunately Bluetooth's RSSI was designed solely for power control and not much effort was put into making it more accurate than it needs to be. The accuracy required to implement power control is very low, and indeed RSSI for power control alone need not even have meaningful units associated with it. Because of Bluetooth's desire to be cheap and simple, the accuracy of the Receive Signal Strength Indicator is not suitable for measuring the actual received power level, making range measurements and therefore position location all but impossible with the current generation of Bluetooth hardware.

The typical environment that Bluetooth operates in also causes problems with trying to use received signal strength to estimate range between devices. With a relatively short wavelength multipath in these environments can cause a significant variation in the received signal strength over a very small distance.

10. References

[1] M. Kayton and W. R. Fried, ed., "Avionics Navigation Systems," New York: John Wiley & Sons, Inc., 1969.

[2] K. W. Bose, "Aviation Electronics," Indianapolis: Howard W. Sams & Co., Inc., 1977

[3] S. Thongthammacharl and H. Olesen, **"**Bluetooth enables in-door mobile location services," Vehicular Technology Conference, Vol. 3 , pp 2023 – 2027, April 2003.

[4] J. Hallberg, M. Nilsson, and K. Synnes, "Bluetooth Positioning," CSEE 2002

[5] Yi-Bing Lin, Hsu-Yung Cheng, Ya-Hsing Cheng, and P. Agrawal, "Implementing Automatic Location Update for Follow-Me Database Using VoIP and Bluetooth Technologies," IEEE Transactions on Computers, Vol. 51, pp. 1154-1168, Oct. 2002

[6] F.J. Gonzalez-Castano and J. Garcia-Reinoso, "Survivable Bluetooth location networks," IEEE International Conference on Communications, Vol. 2, pp. 1014-1018, May 2003

[7] D. Paiidya, R. Jain, and E. Lupu, "lndoor location estimation using multiple wireless technologies," IEEE International Symposium on Persona1,lndoor and Mobile Radio Communication, Proceedings, Vol.3, pp. 2208-2212, 2003

[8] W. Zhuang, Chi-Hsiang Yeh, O. Droegehorn, C.-T. Toh, and H. R. Arabnia, "An indoor Bluetooth-based positioning system: concept, Implementation and experimental evaluation," International Conference on Wireless Networks, June 2003

[9] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless LANs," IEEE International Symposium Personal, Indoor and Mobile Radio Communications, Vol. 2, pp. 720-724, Sept. 2002

[10] "Specification of the Bluetooth System: Version 1.1," Bluetooth Special Interest Group, Feb. 2001

[11] "BlueCore01 Transmit Power Control Application Note AN051," Cambridge Silicon Radio, May 2001

[12] "BlueCore01 Persistent Store Key Settings AN102," Cambridge Silicon Radio, Sept. 2001

[13] T. S. Rappaport, Wireless Communications: Principles and Practice 2nd edition, Prentice Hall PTR, Upper Saddle River, NJ, 2002

[14] Jaap C. Haartsen and Stefan Zürbes, "Bluetooth voice and data performance in 802.11 DS WLAN environment," SIG publication, pp. 1-13.

[15] E. D. Kaplan, ed., "Understanding GPS Principles and Applications," Boston: Artech House Publishers, 1996

[16] R. Christ, "Application and Performance of Personnel Tracking Systems," International Carnahan Conference on Security Technology, pp 120-128, Oct. 1996

Appendix A: Casira Cable Attenuation vs. Reported RSSI Plots

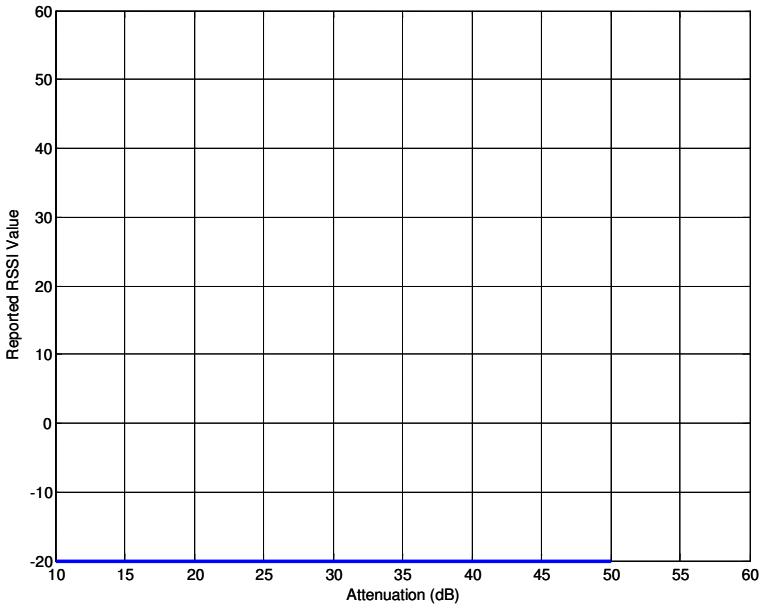


Figure 47. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=150.

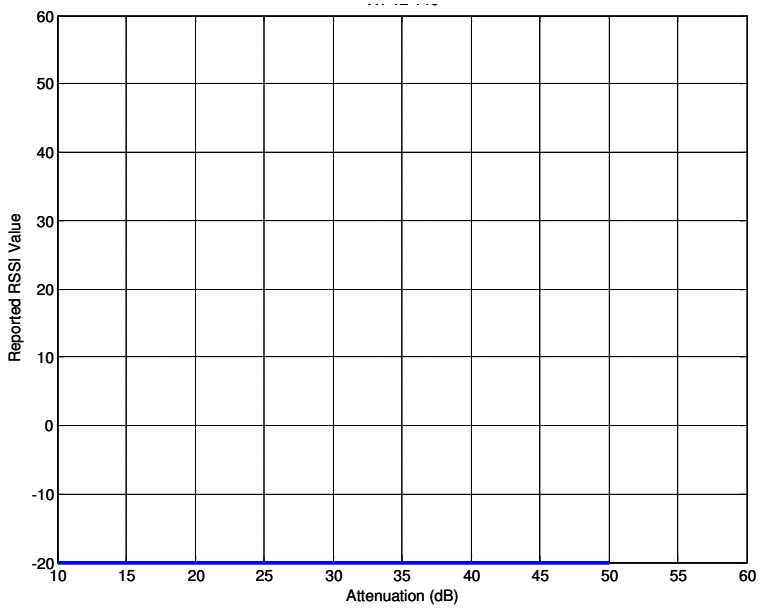


Figure 48. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=140.

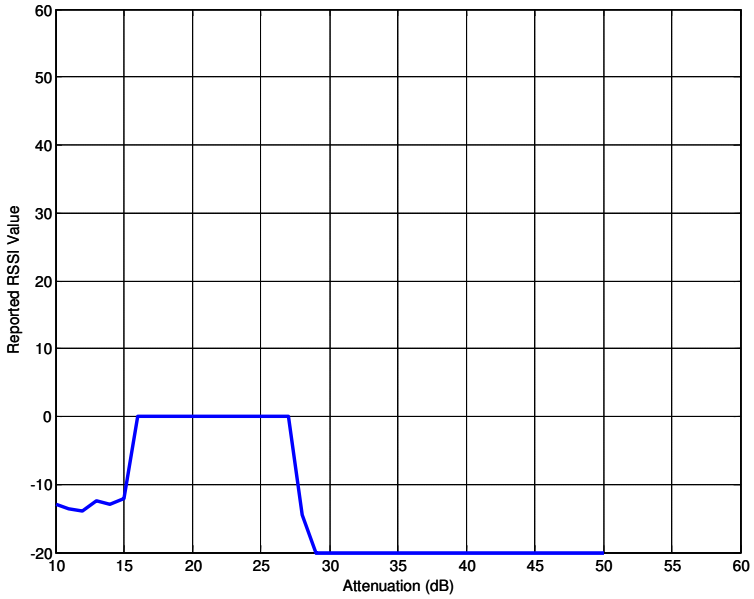


Figure 49. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=130.

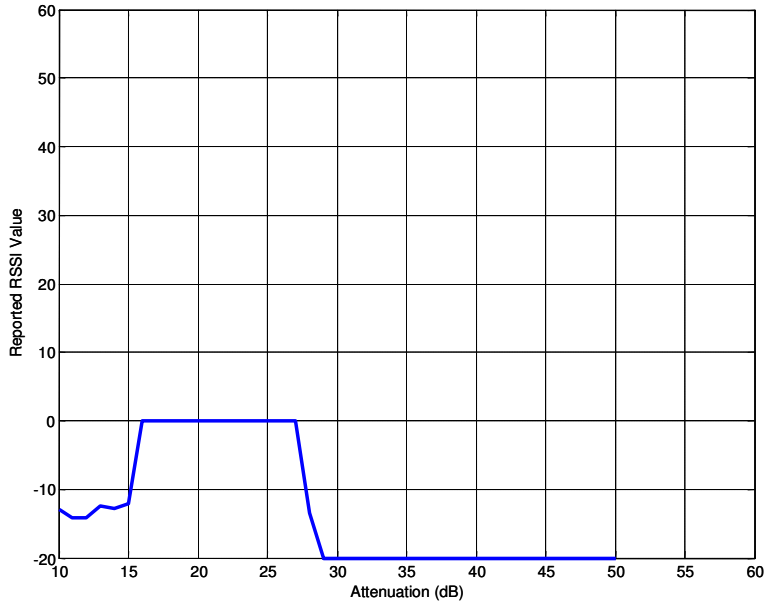


Figure 50. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=120.

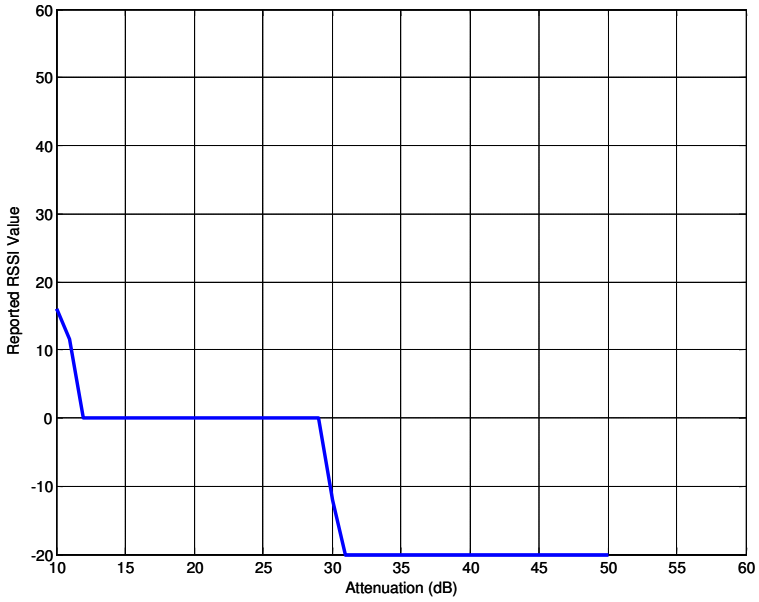


Figure 51. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=110.

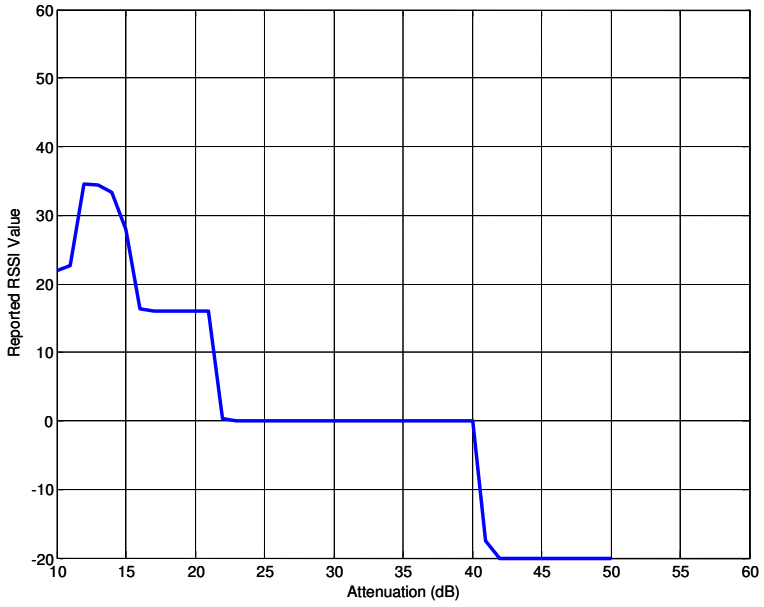


Figure 52. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=100.

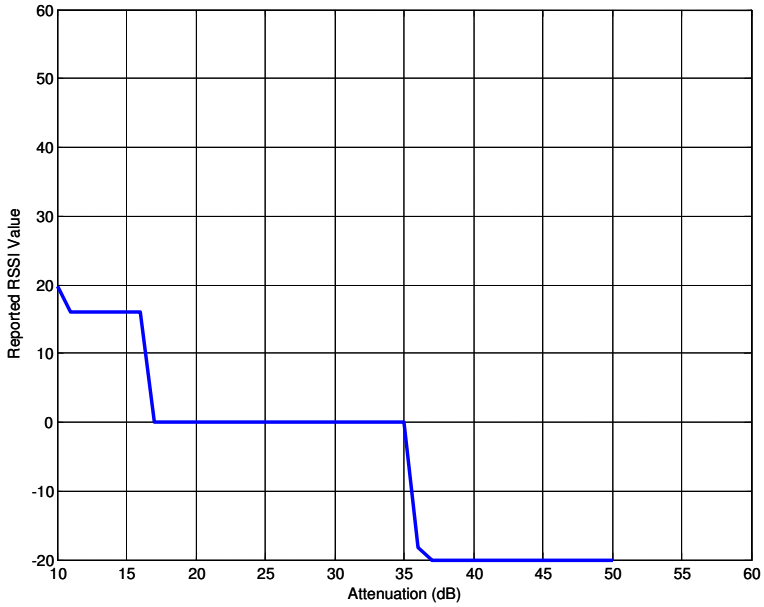


Figure 53. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=90.

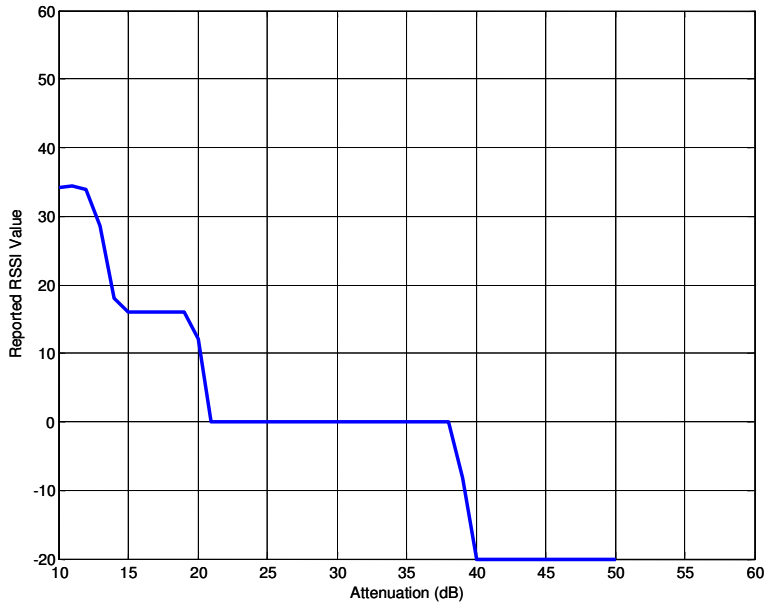


Figure 54. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

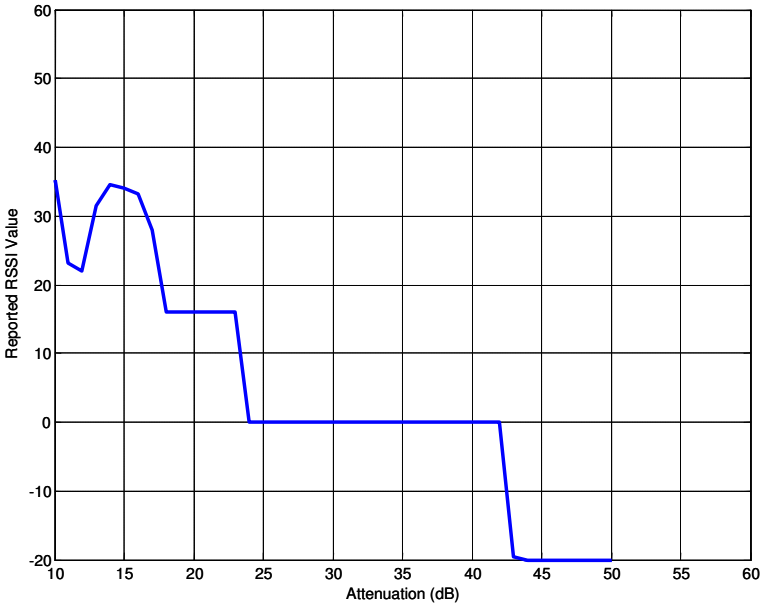


Figure 55. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=70.

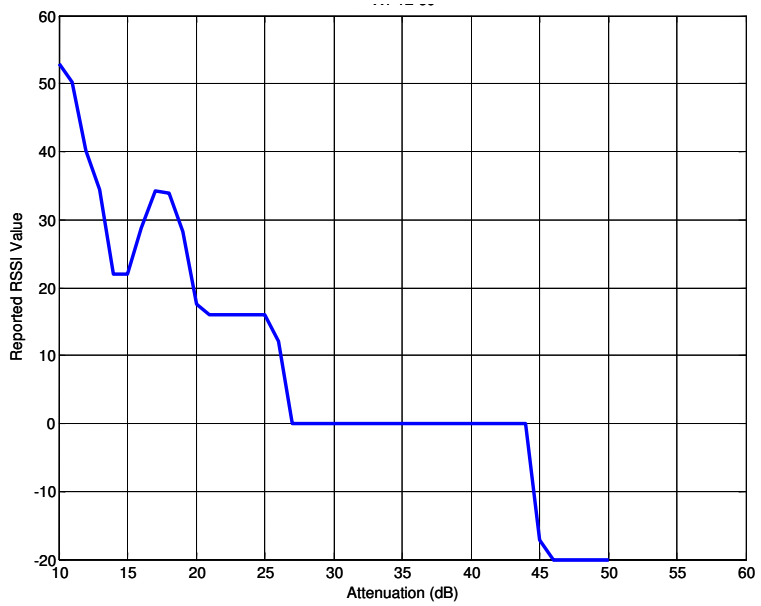


Figure 56. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=60.

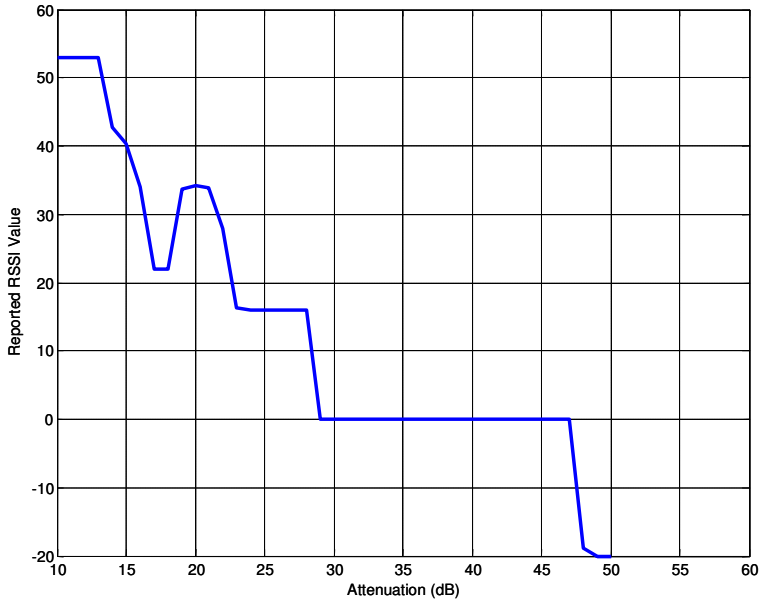


Figure 57. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=50.

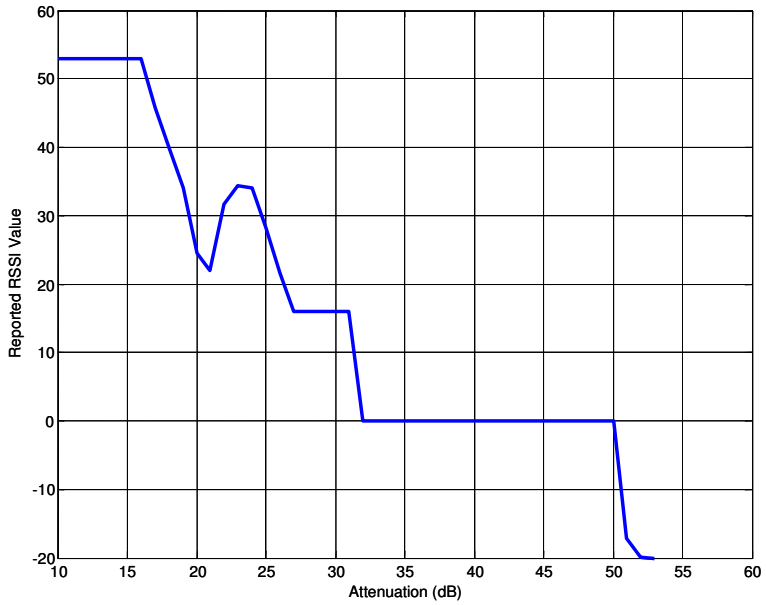


Figure 58. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=40.

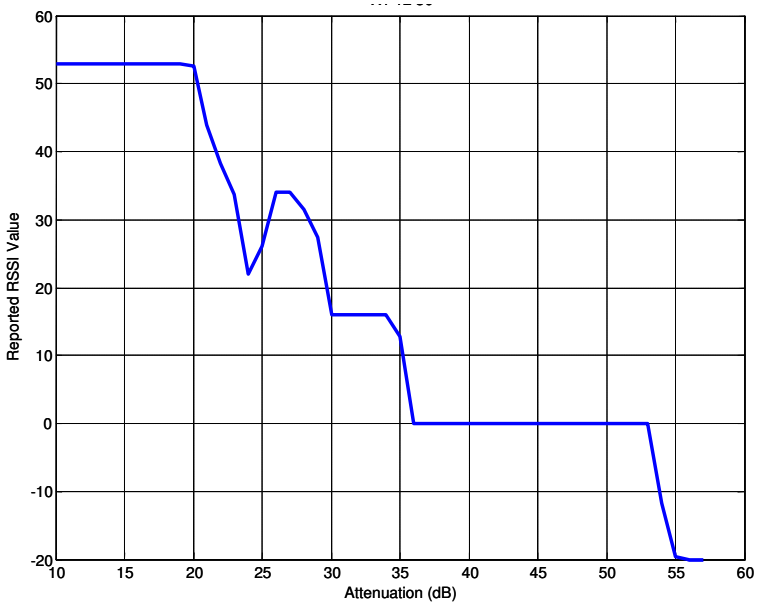


Figure 59. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=30.

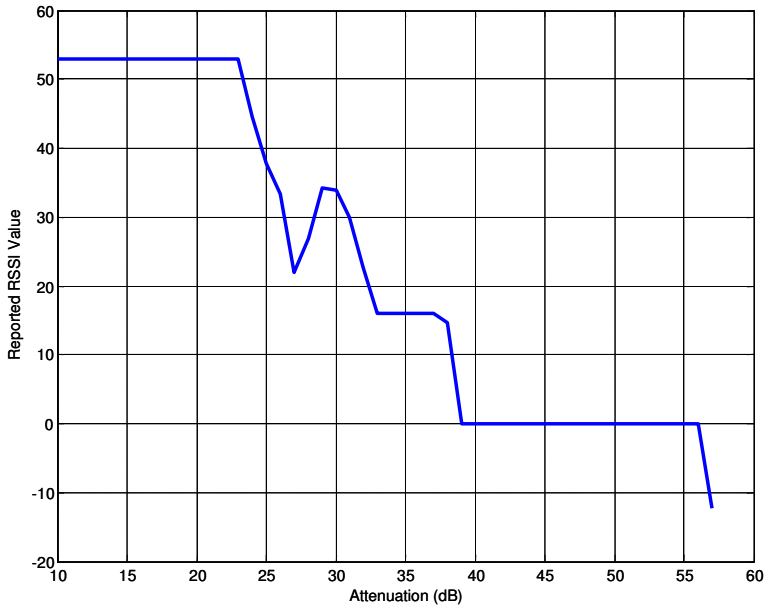


Figure 60. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=20.

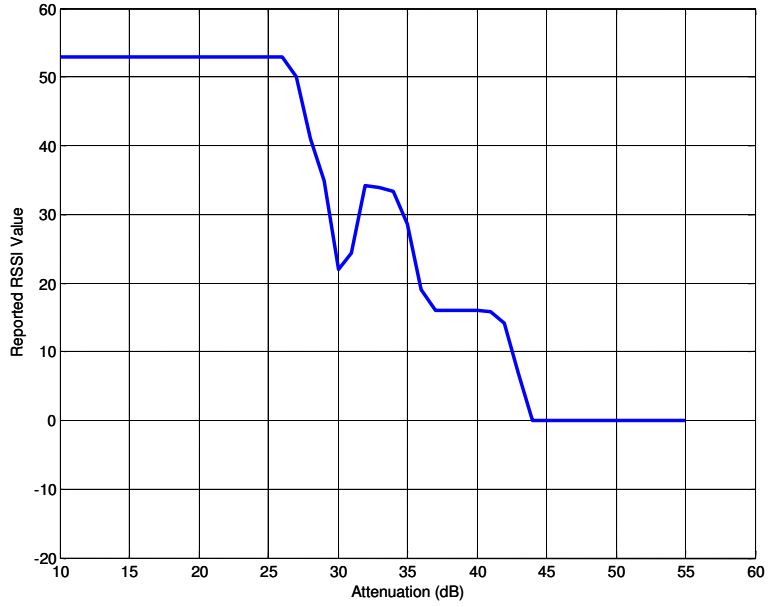


Figure 61. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=10.

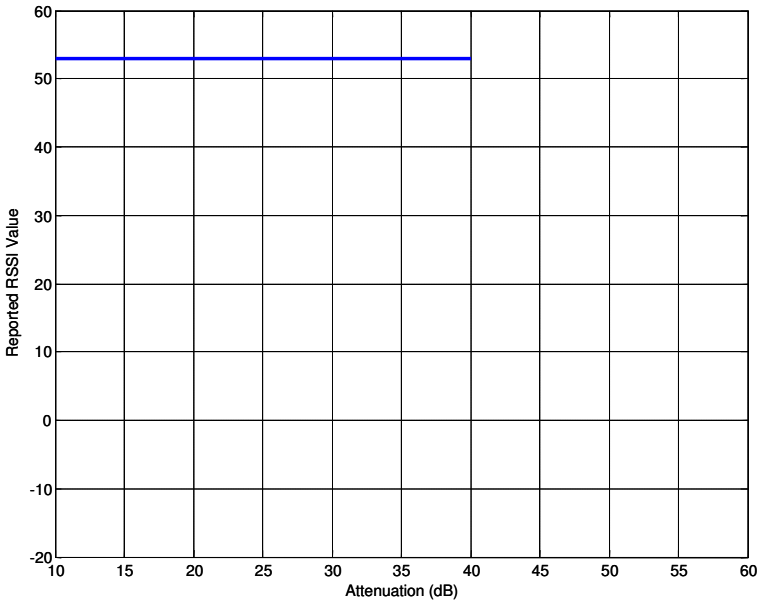


Figure 62. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=0.

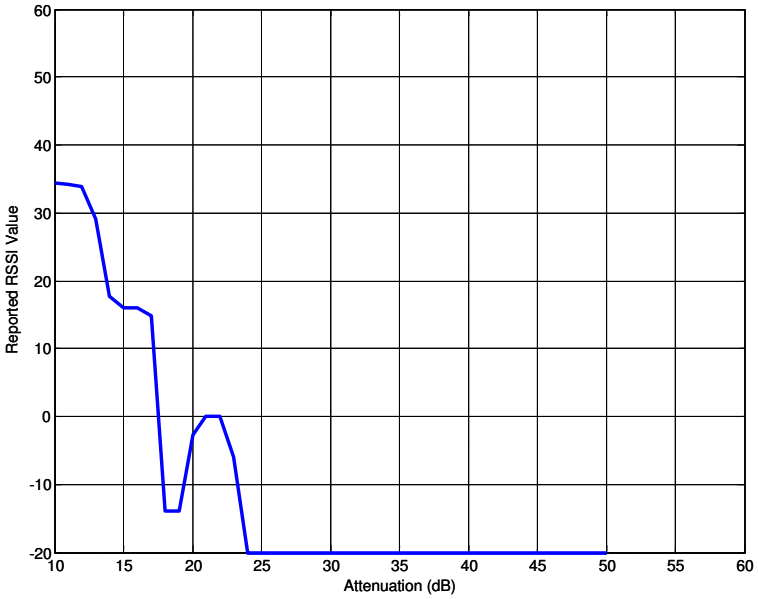


Figure 63. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=12 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

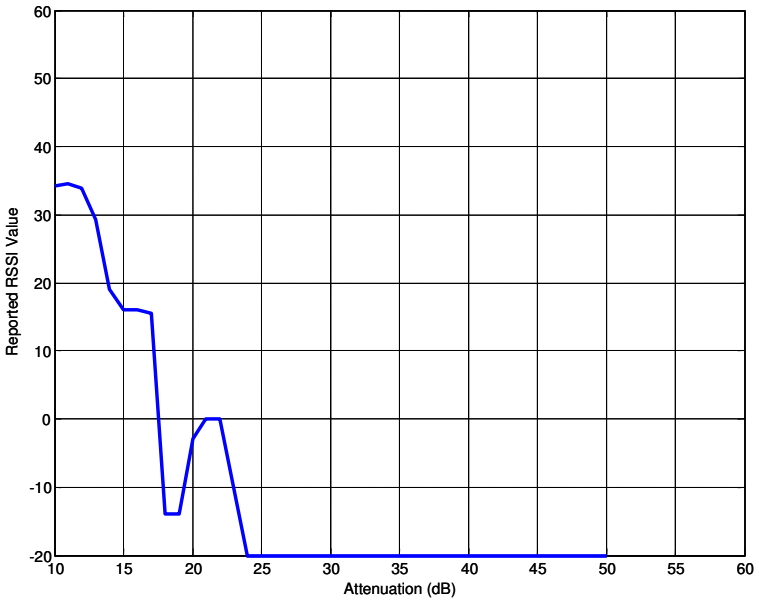


Figure 64. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=11 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

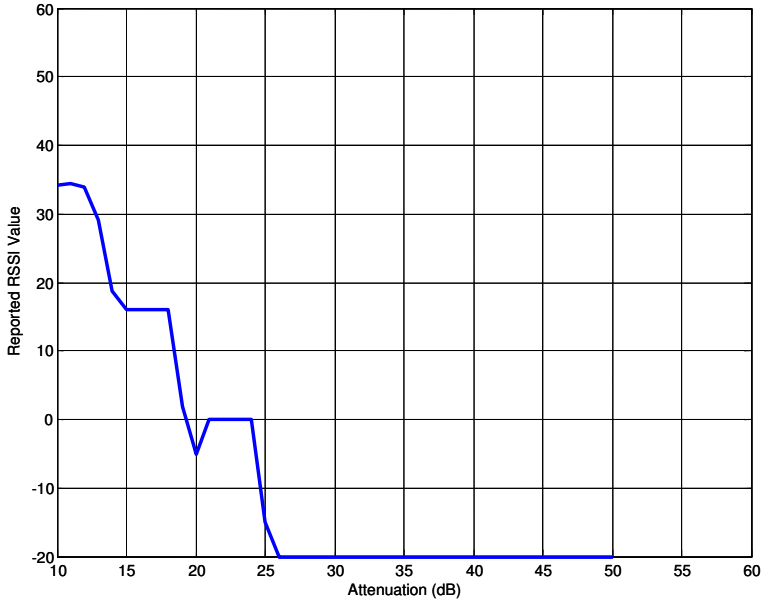


Figure 65. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=10 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

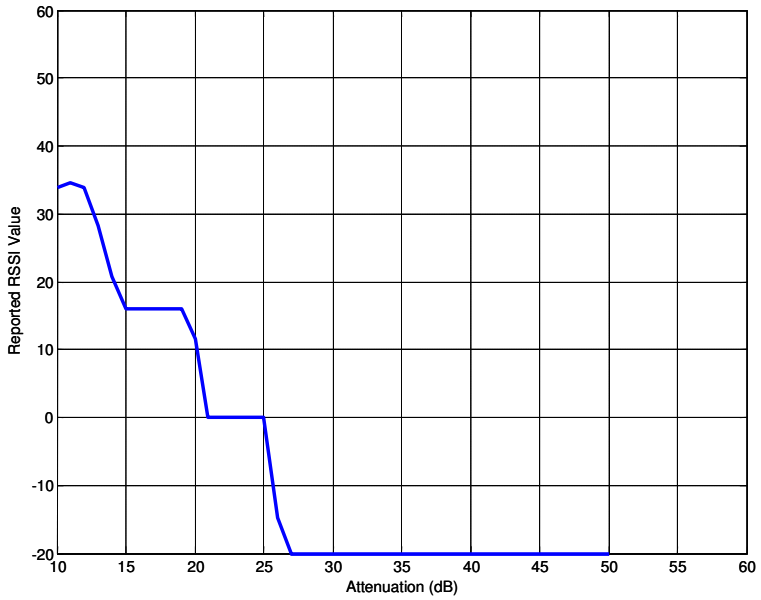


Figure 66. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=9 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

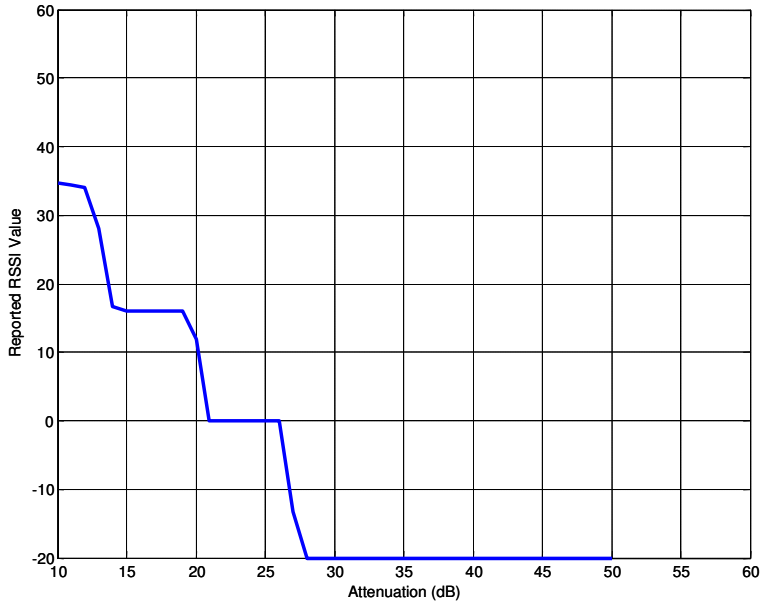


Figure 67. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=8 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

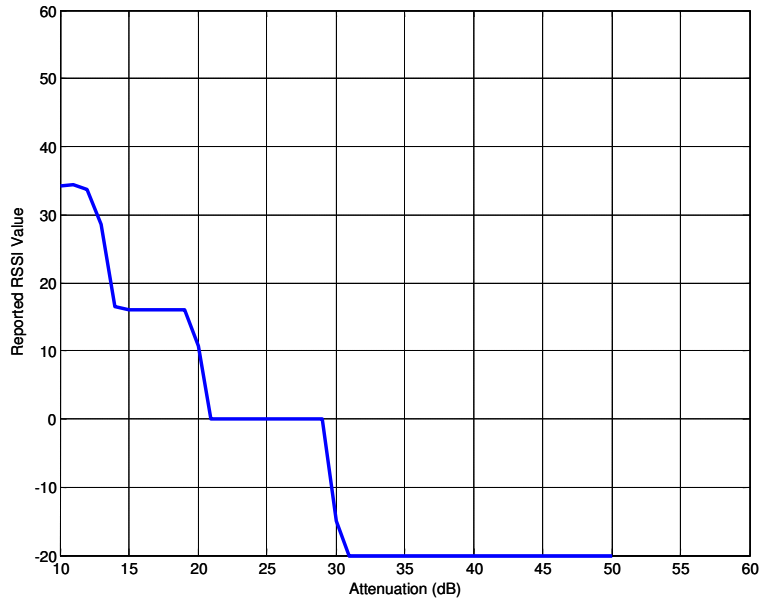


Figure 68. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=7 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

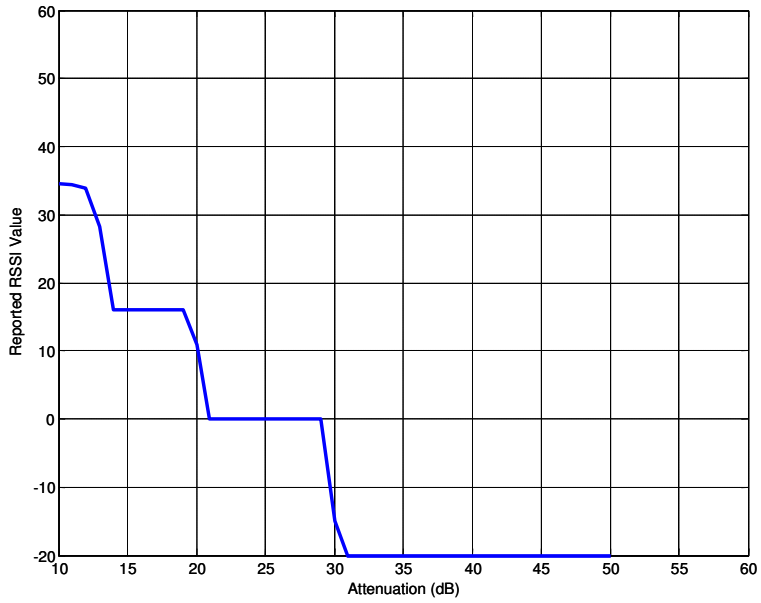


Figure 69. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=6 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

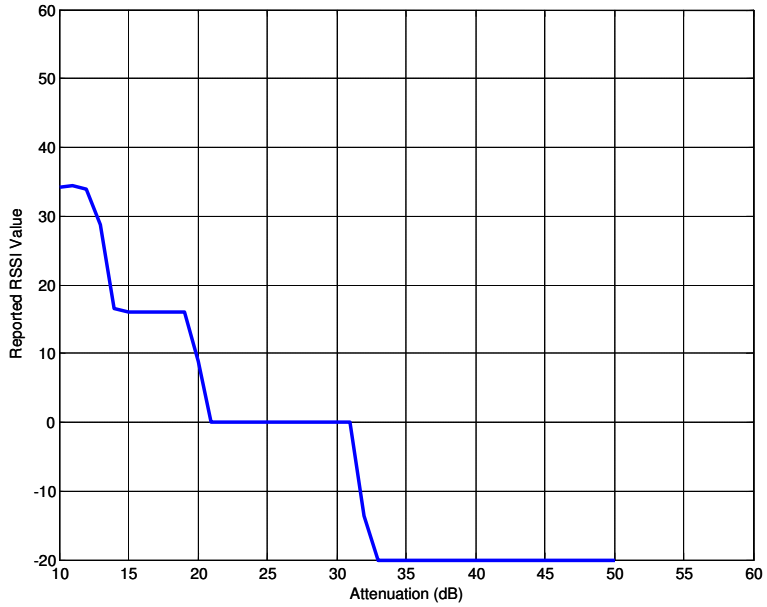


Figure 70. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=5 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

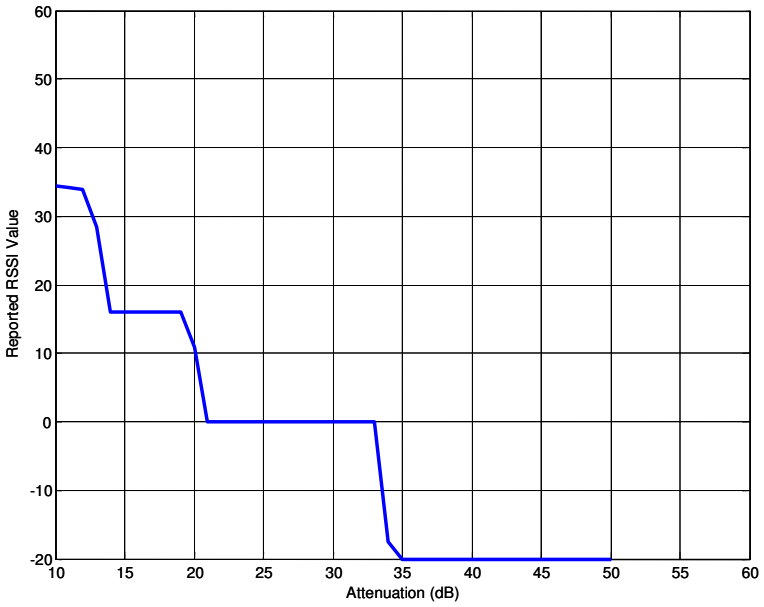


Figure 71. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=4 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

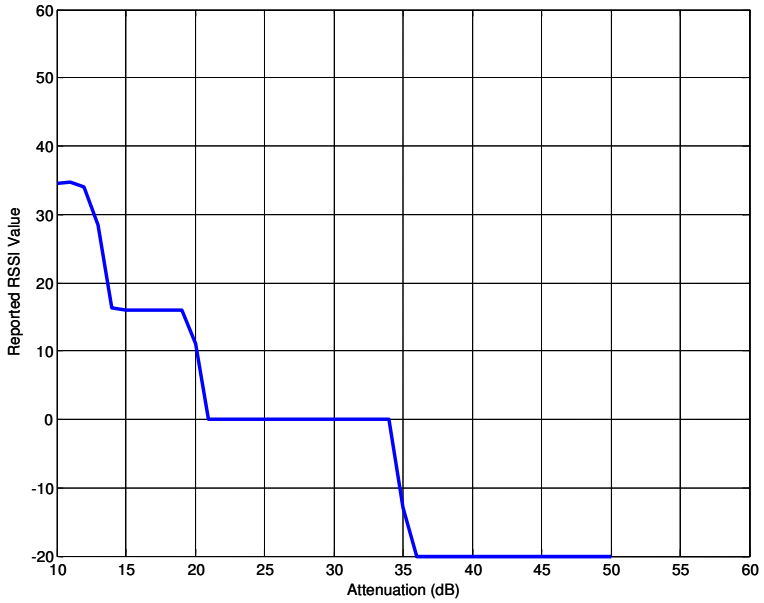


Figure 72. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=3 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

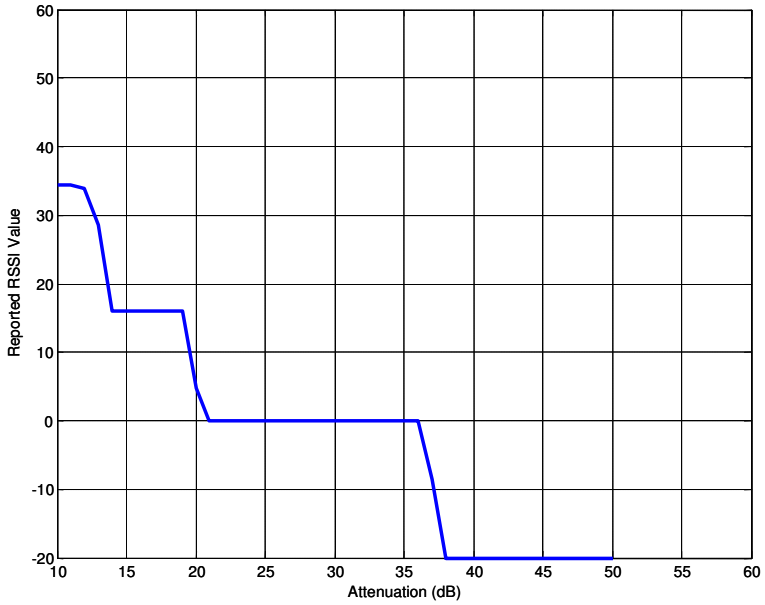


Figure 73. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=2 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=12 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

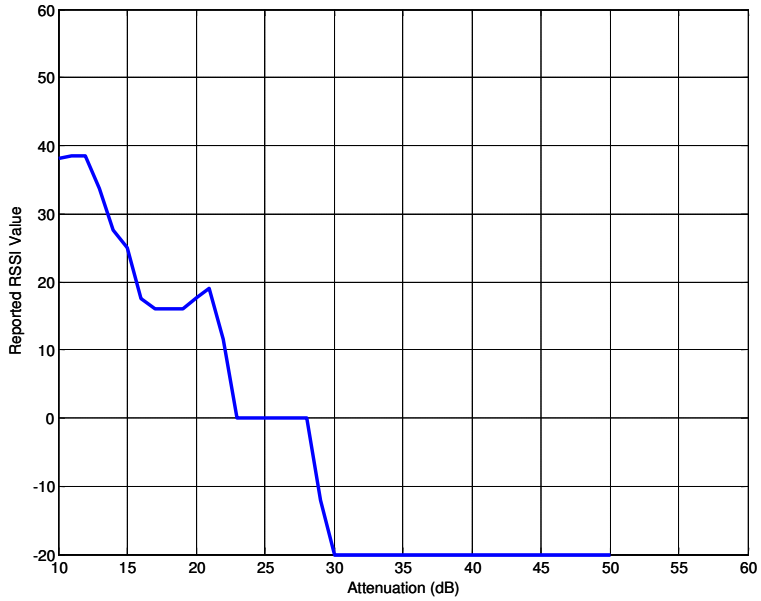


Figure 74. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=11 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

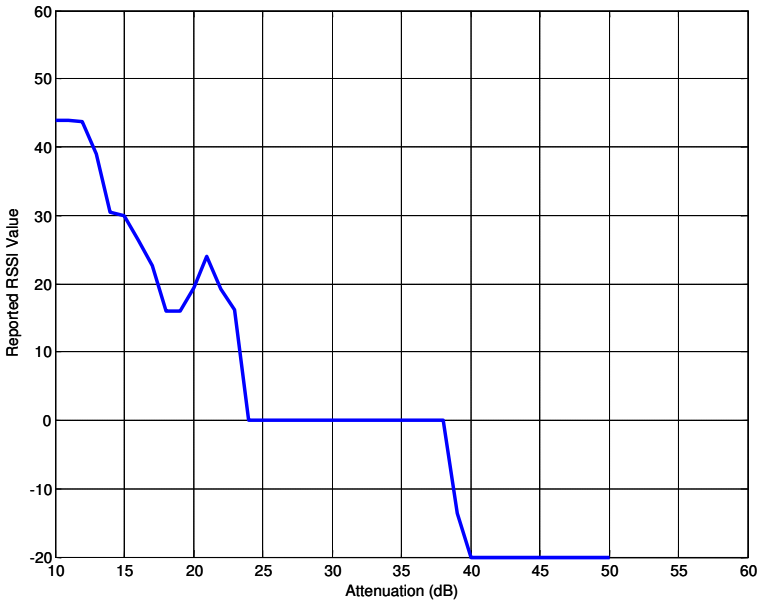


Figure 75. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=10 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

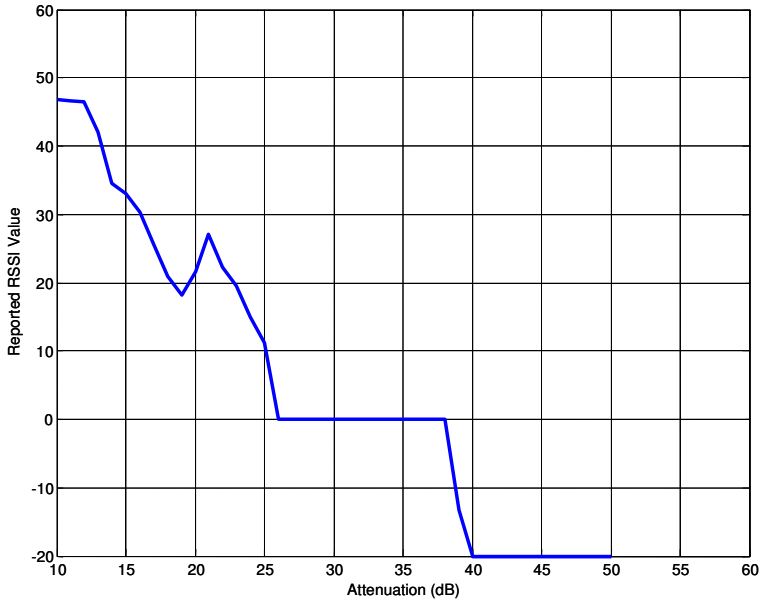


Figure 76. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=9 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

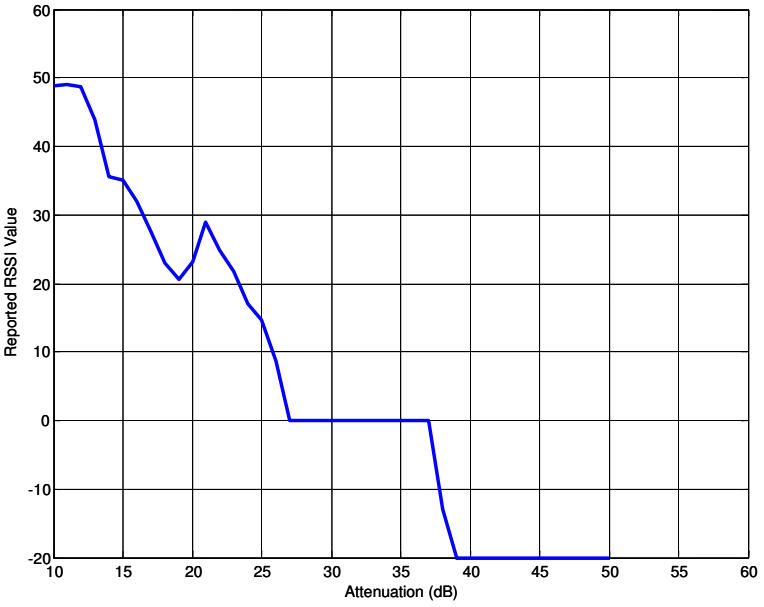


Figure 77. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=8 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

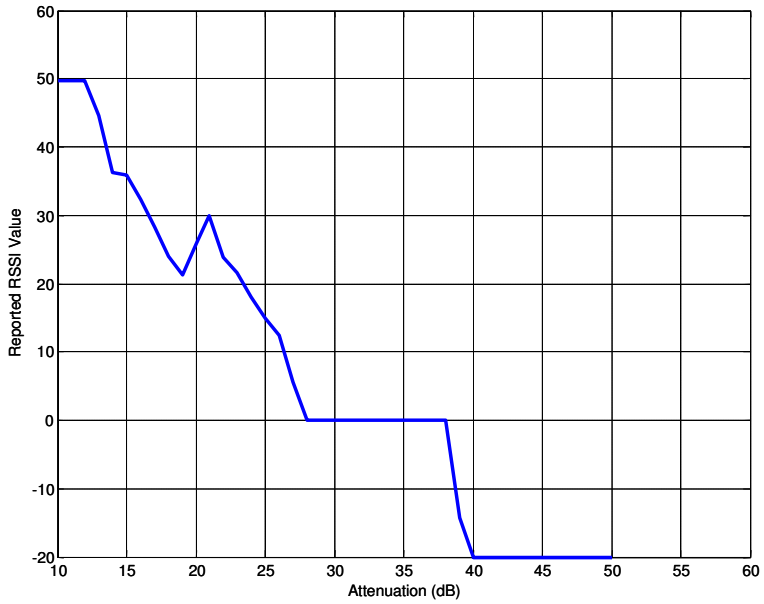


Figure 78. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=7 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

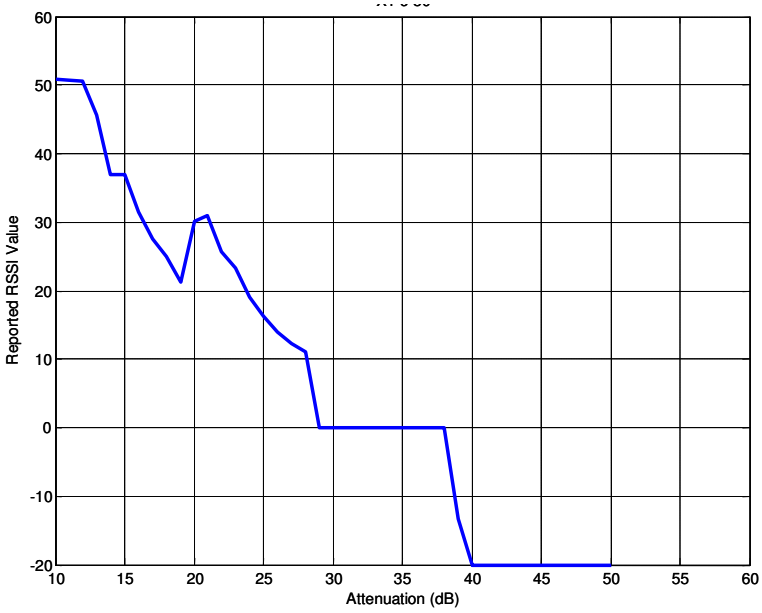


Figure 79. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=6 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

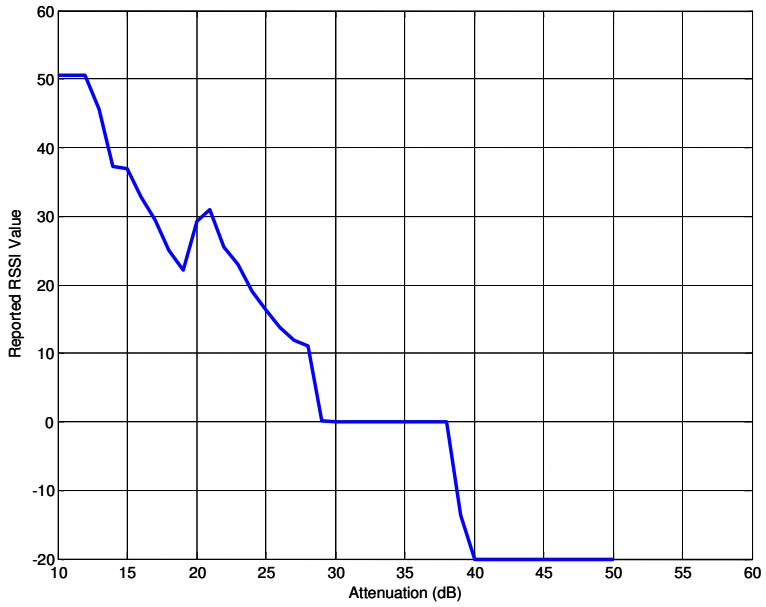


Figure 80. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=5 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

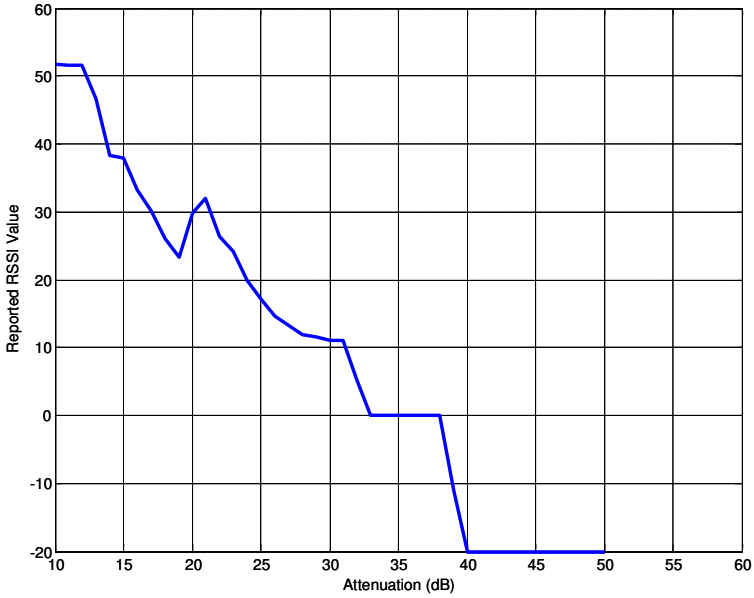


Figure 81. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

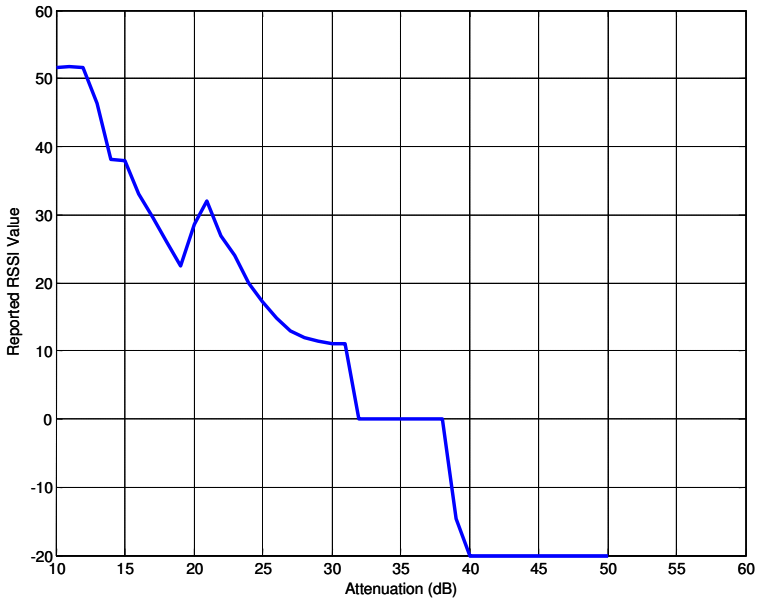


Figure 82. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=3 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

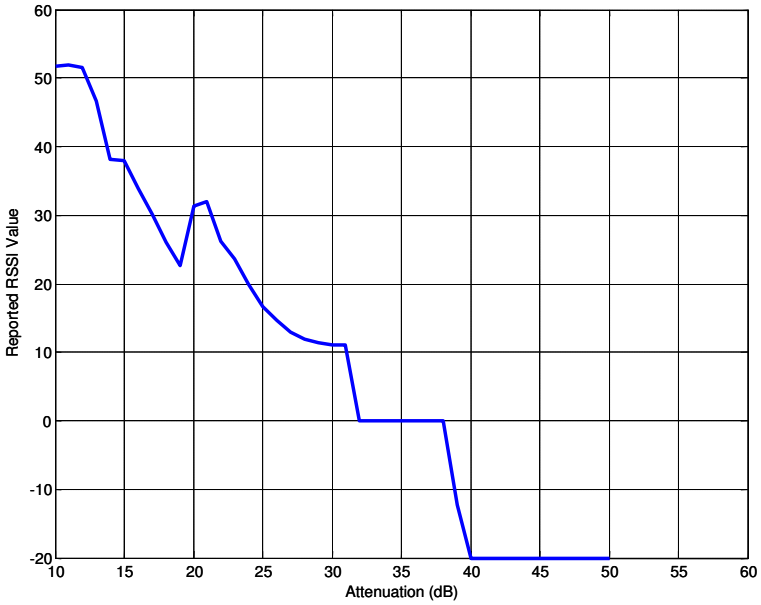


Figure 83. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=2 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

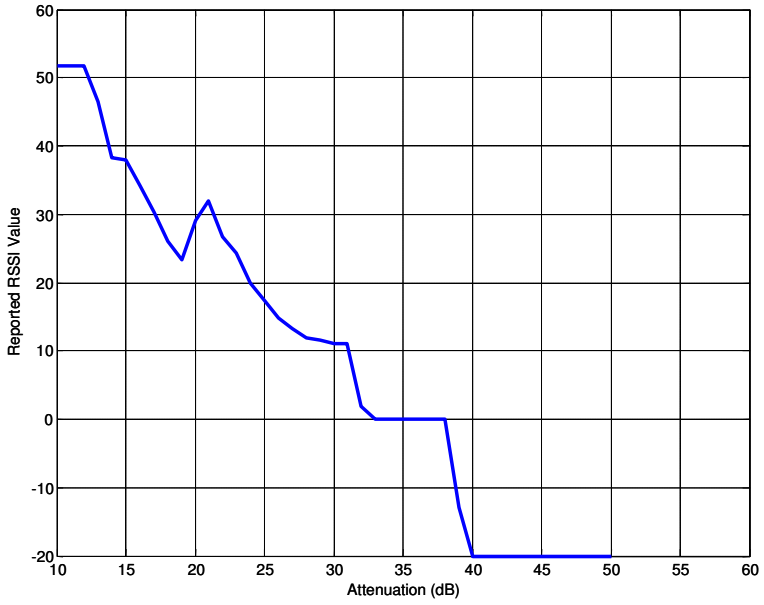


Figure 84. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=1 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

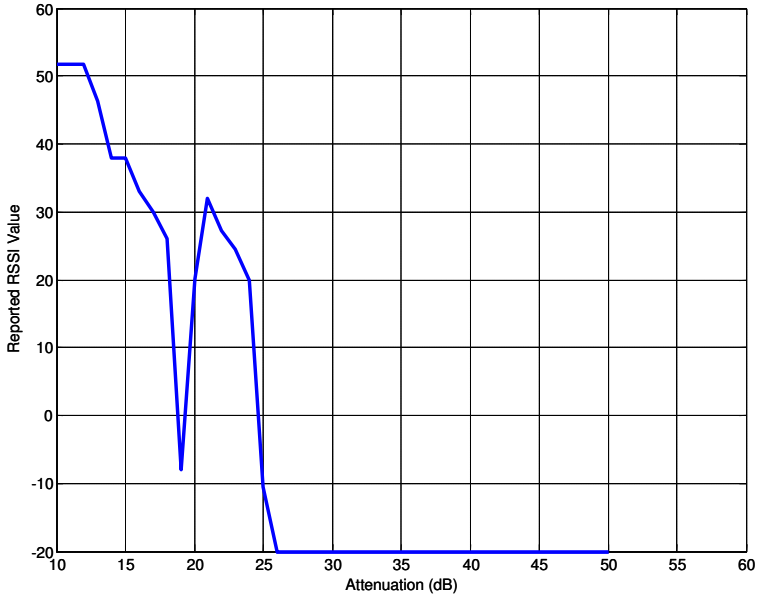


Figure 85. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=10 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

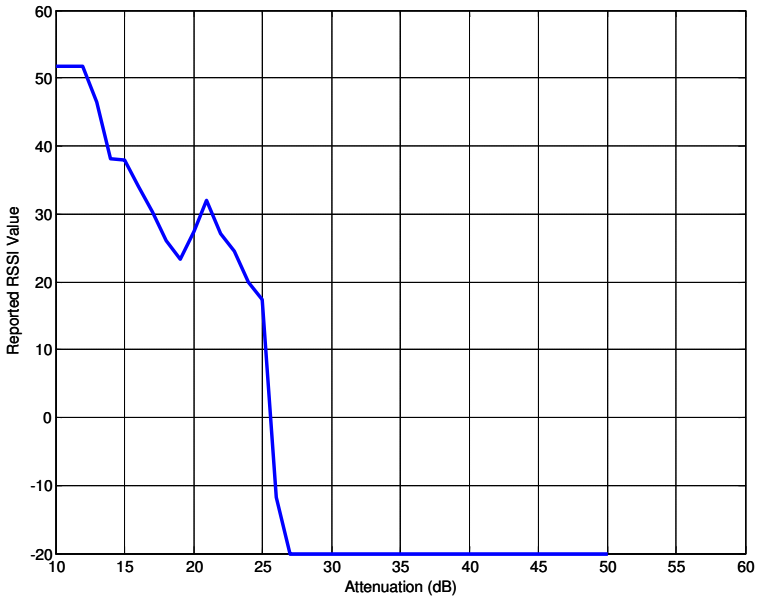


Figure 86. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=9 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

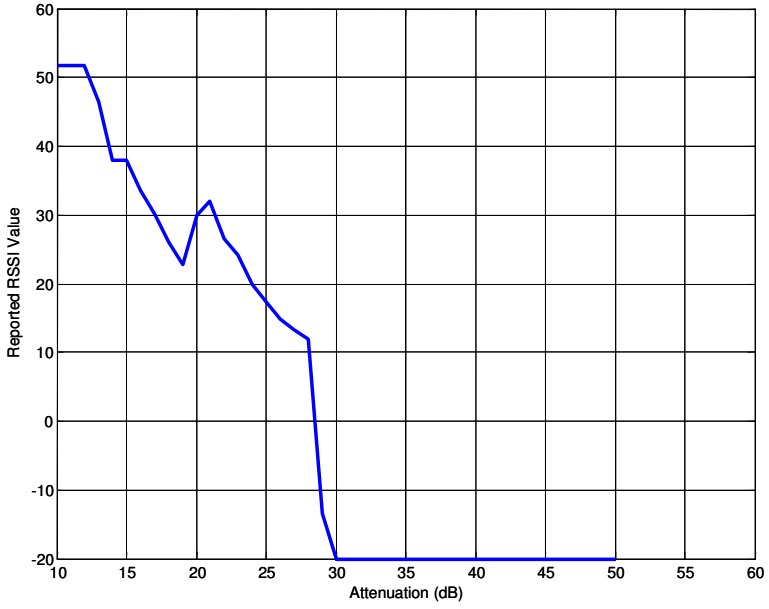


Figure 87. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=8 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

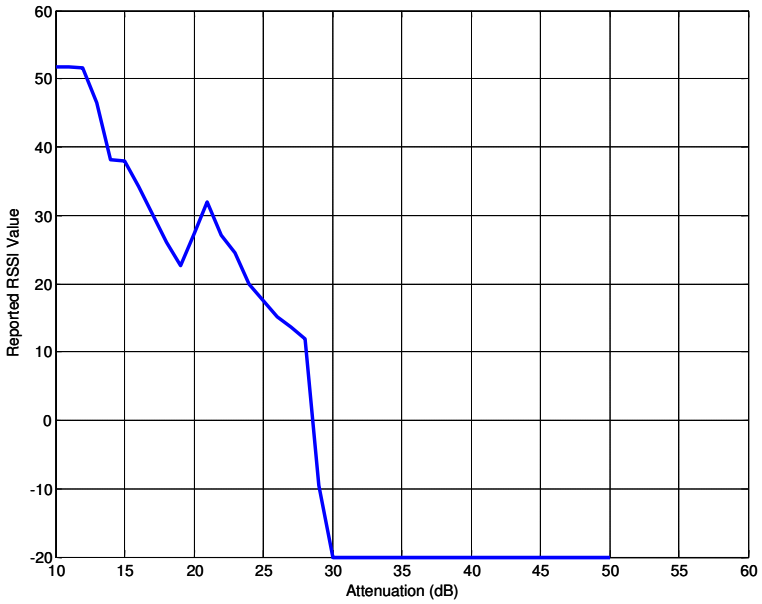


Figure 88. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=7 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

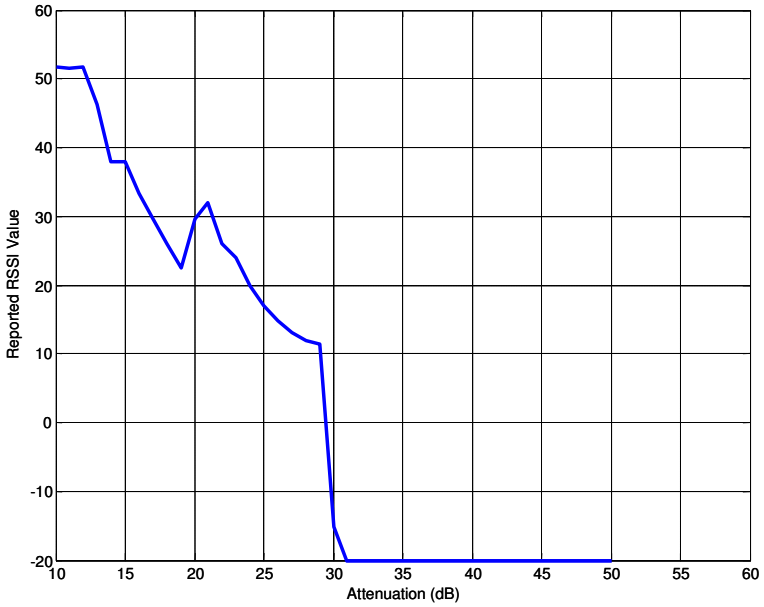


Figure 89. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=6 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

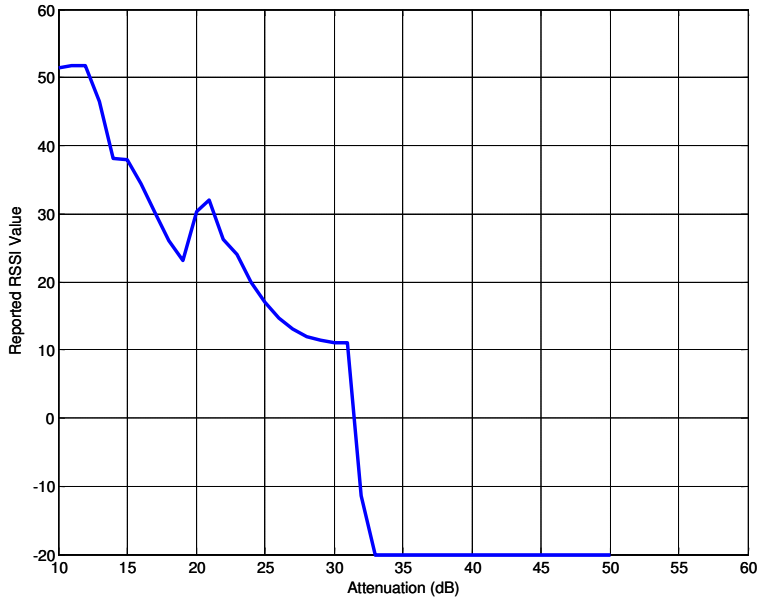


Figure 90. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=5 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

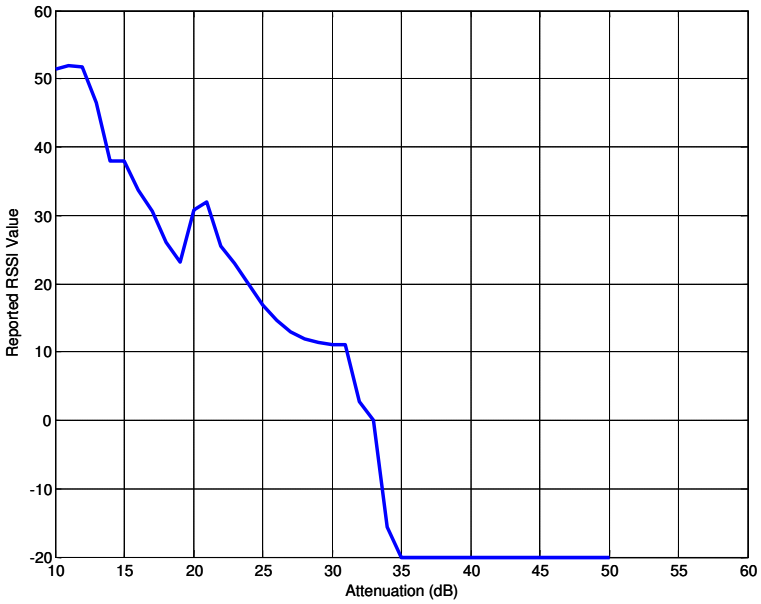


Figure 91. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=4 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

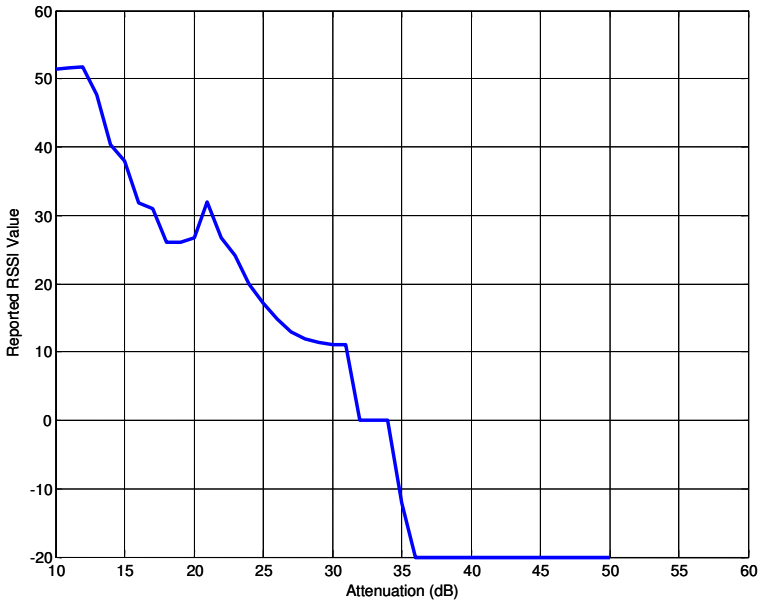


Figure 92. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=3 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

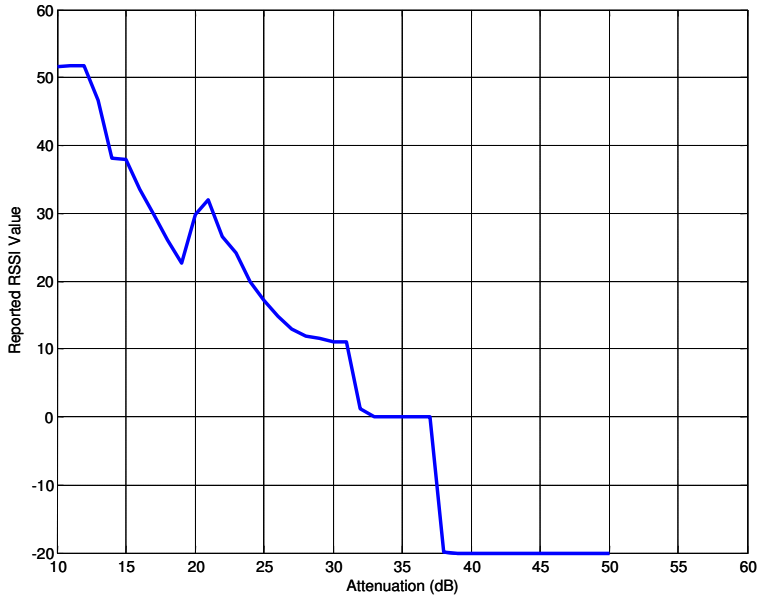


Figure 93. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=2 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

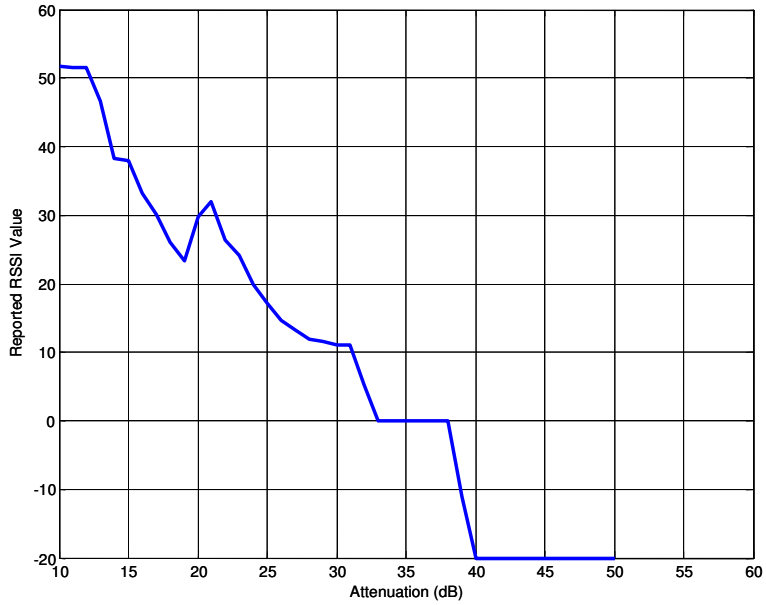


Figure 94. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=1 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

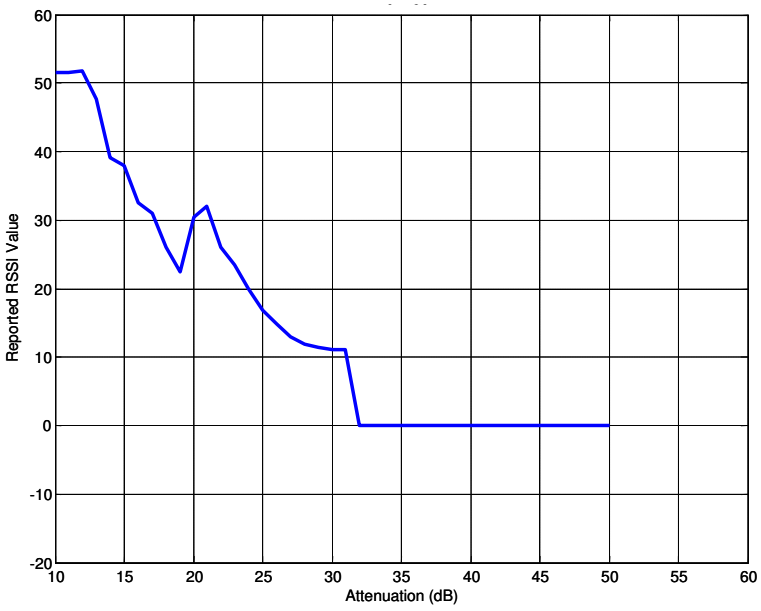


Figure 95. Cable Attenuation vs. Reported RSSI with PSKEY_LC_ATTEN_GOLDEN_RANGE_MINIMUM=0 PSKEY_LC_ATTEN_GOLDEN_RANGE_MAXIMUM=4 PSKEY_LC_RSSI_GOLDEN_RANGE=80.

Appendix B: Range Measurement Plots

Indoor Measurements

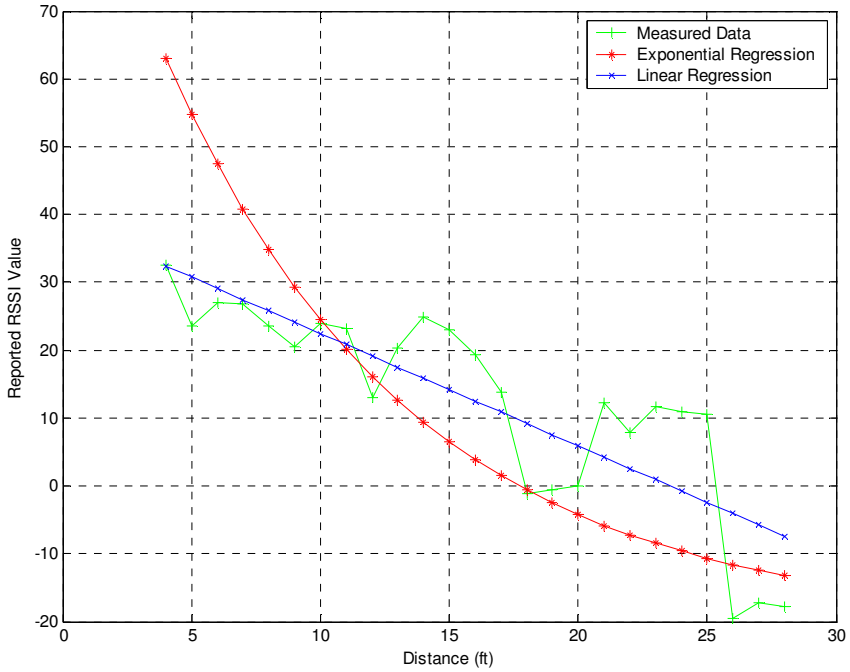


Figure 96. Indoor measurement set #1.

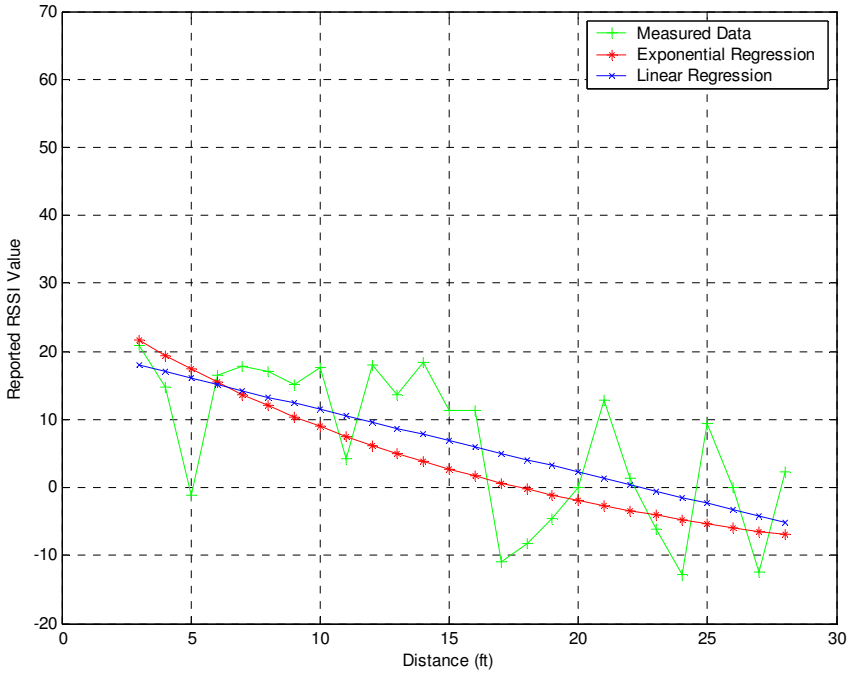


Figure 97. Indoor measurement set #2.

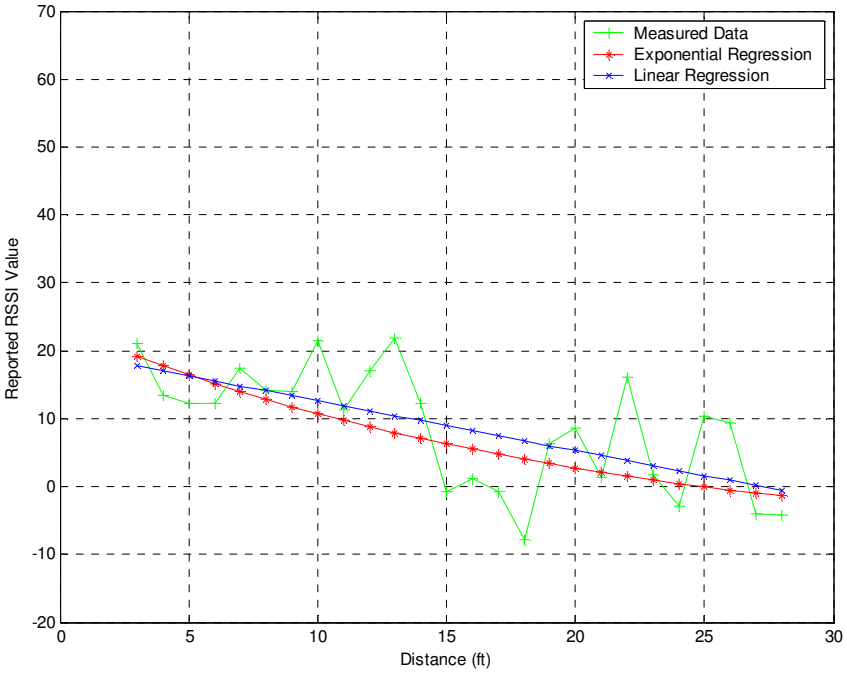


Figure 98. Indoor measurement set #3.

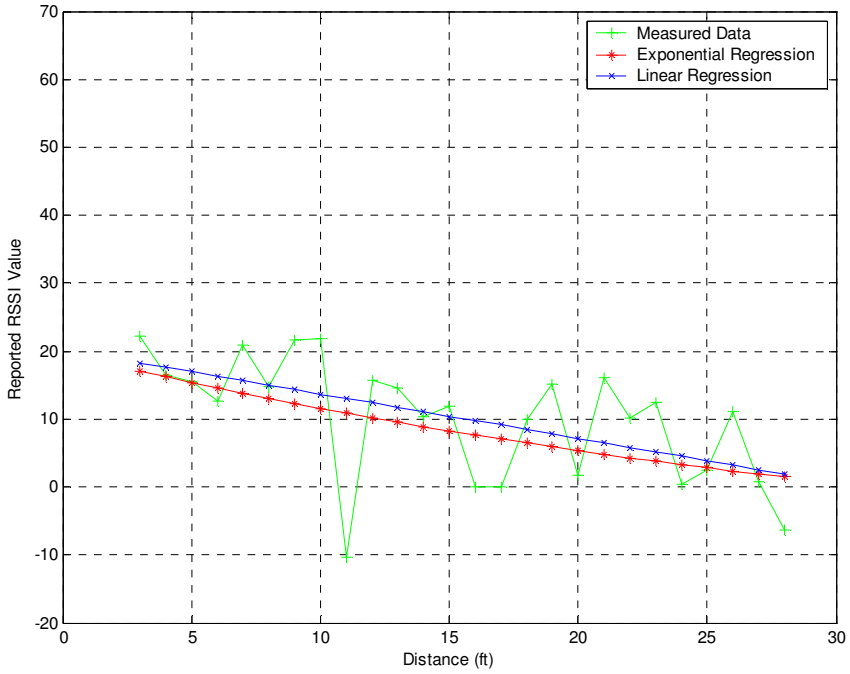


Figure 99. Indoor measurement set #4.

Outdoor Measurements

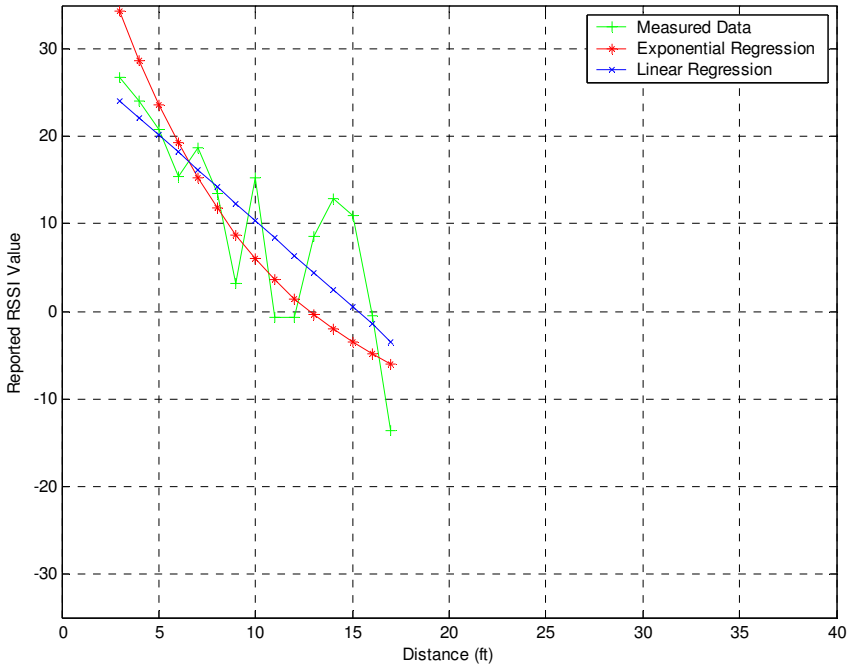


Figure 100. Outdoor measurement set #1.

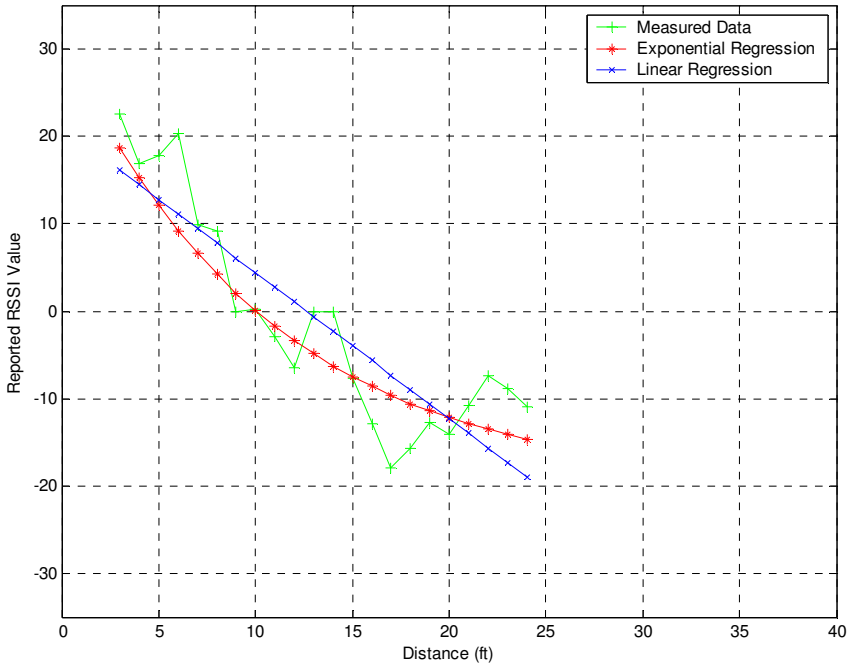


Figure 101. Outdoor measurement set #2.

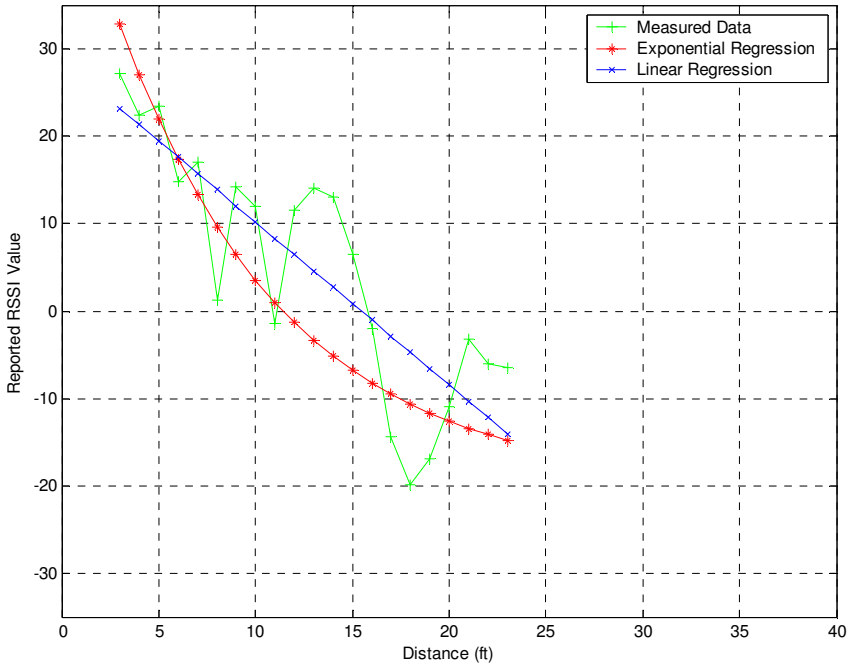


Figure 102. Outdoor measurement set #3.

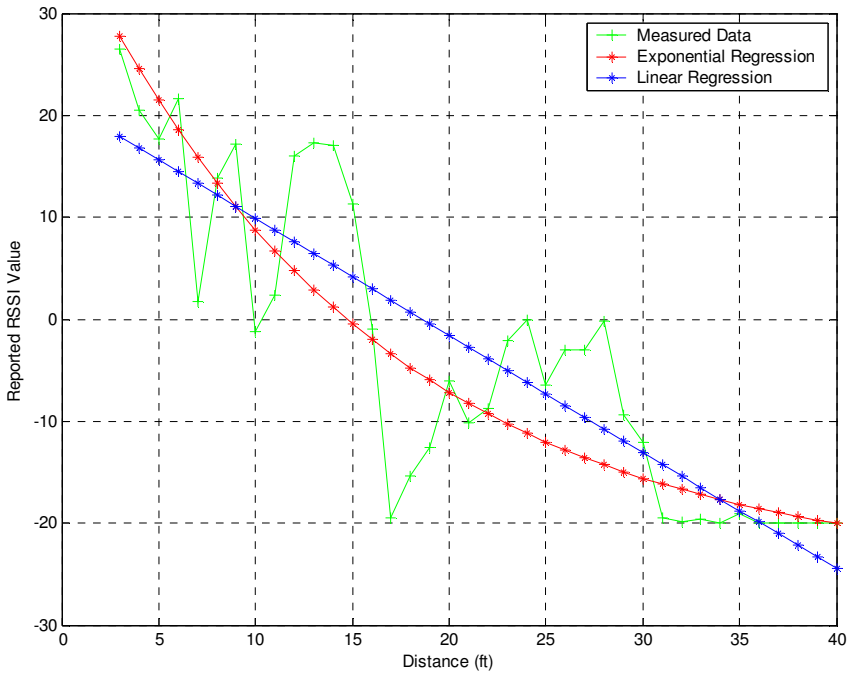


Figure 103. Outdoor measurement set #4.

Range Estimation Measurements

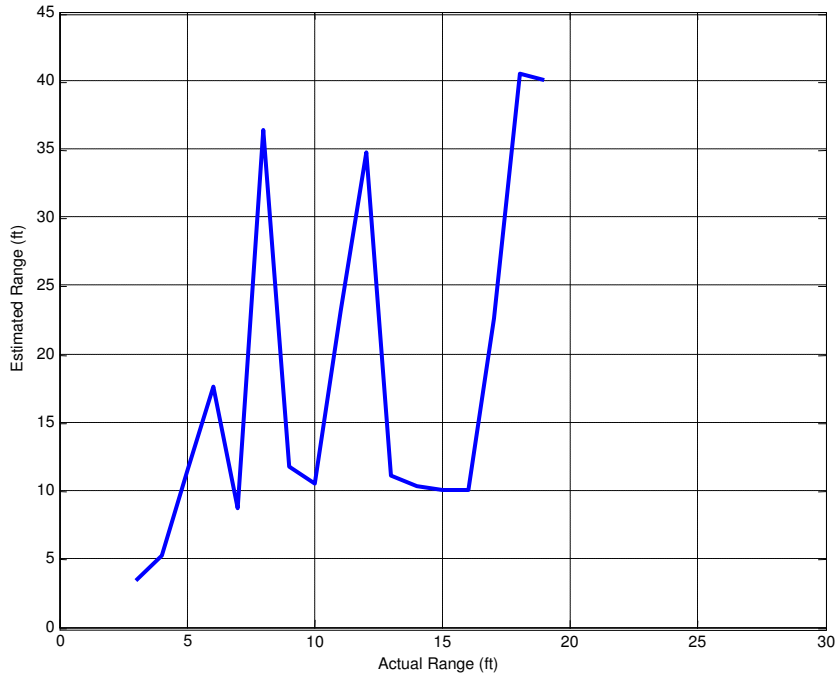


Figure 104. Outdoor range estimation set #1.

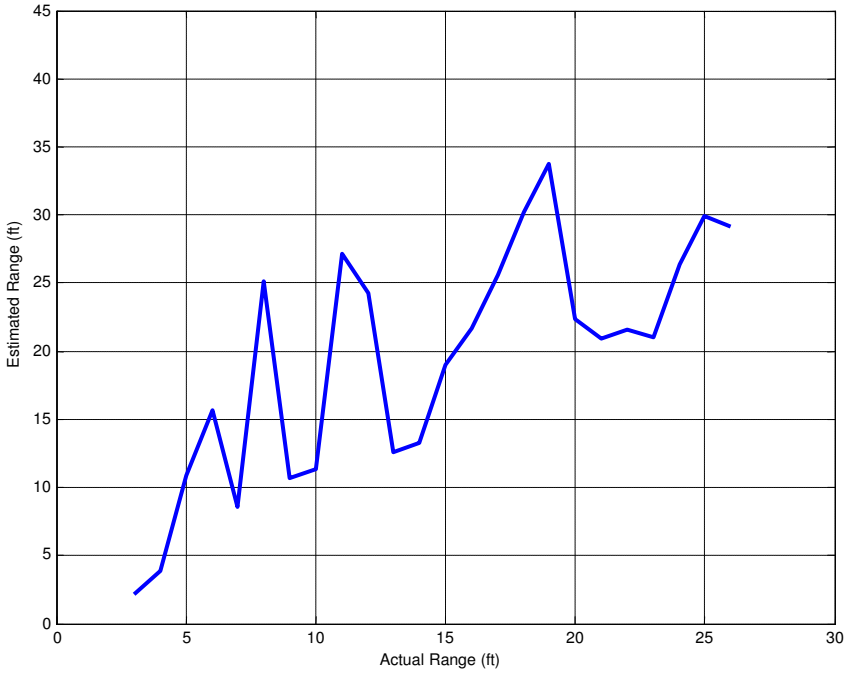


Figure 105. Outdoor range estimation set #2.

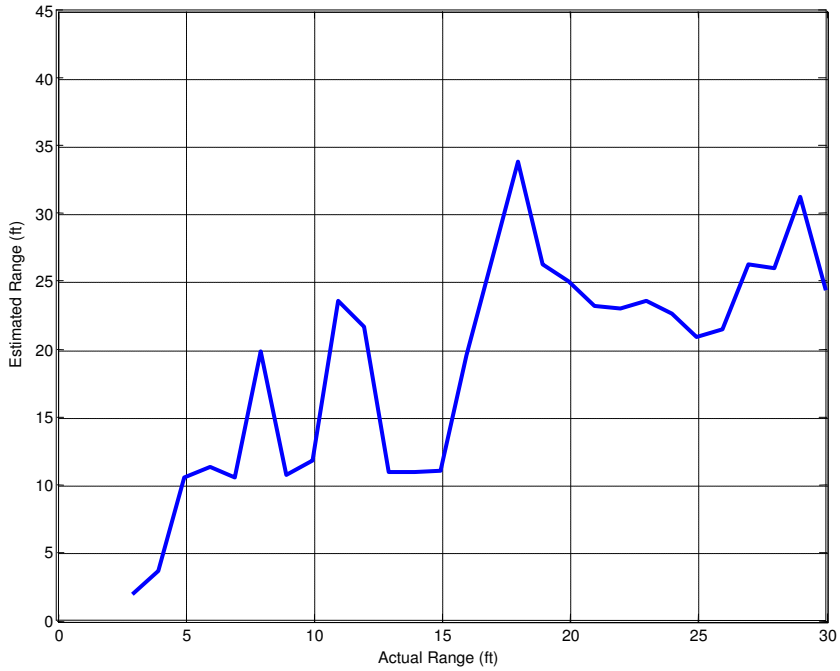


Figure 106. Outdoor range estimation set #3.

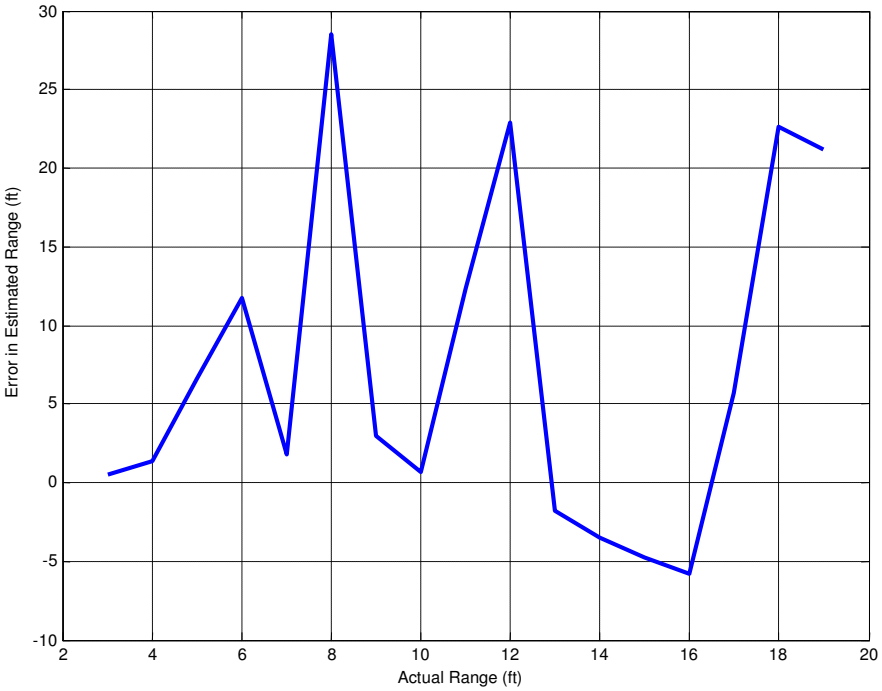


Figure 107. Error in Outdoor range estimation set #1.

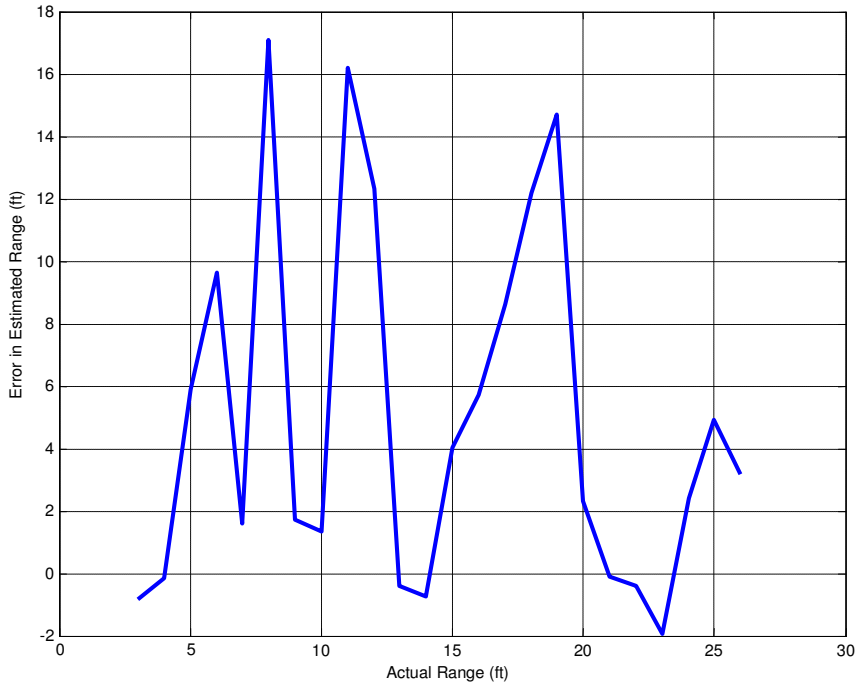


Figure 108. Error in Outdoor range estimation set #2.

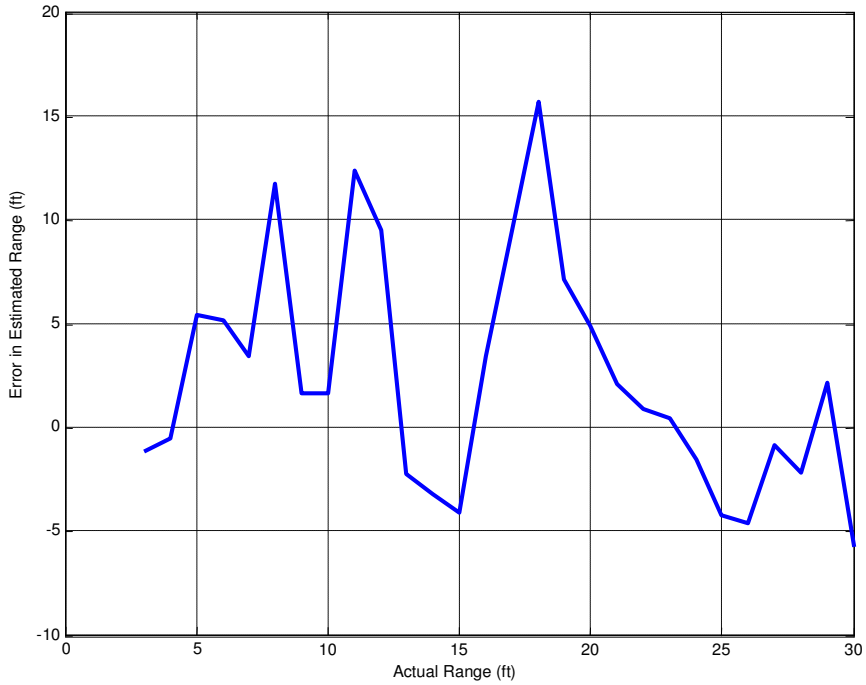


Figure 109. Error in Outdoor range estimation set #3.

Vita

Name: Timothy M. Bielawa

Date and Place of Birth: January 9, 1980
Boston, Massachusetts

Education:

High School: Alvirne High School
Hudson, NH
Graduated in June 1998

College: Virginia Polytechnic Institute and State University
Blacksburg, VA
B.S.E.E., May 2002
M.S.E.E., July 2005

Publications:

"A parametric study of time-domain characteristics of possible UWB antenna architectures," Licul, S.; Noronha, J.A.N.; Davis, W.A.; Sweeney, D.G.; Anderson, C.R.; Bielawa, T.M.; Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th Volume 5, 6-9 Oct. 2003 Page(s):3110 - 3114 Vol.5

"Designing Antennas For UWB Systems," Joseph A. N. Noronha, Timothy Bielawa, Christopher R. Anderson, Dennis G. Sweeney, Stanislav Licul, William A. Davis, Microwaves & RF, June 2003